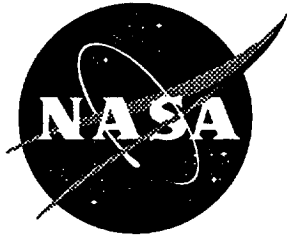


NASA Contractor Report 201679

1N-01  
023273



# The ASAC Flight Segment and Network Cost Models

Bruce J. Kaplan, David A. Lee, Nusrat Retina, and Earl R. Wingrove III  
*Logistics Management Institute, McLean, Virginia*

Brett Malone, Stephen G. Hall, and Scott A. Houser  
*Phoenix Integration, Blacksburg, Virginia*

Contract NAS2-14361

April 1997

National Aeronautics and  
Space Administration  
Langley Research Center  
Hampton, Virginia 23681-0001



# Contents

---

The ASAC Flight Segment and Network Cost Models .....	1
SUMMARY .....	1
INTRODUCTION .....	3
NASA's Role in Promoting Aviation Technology .....	3
NASA's Research Objective .....	3
Goal of the ASAC Project: Identifying and Evaluating Promising Technologies .....	4
Technology Impacts on Airline Costs .....	4
Overview of the ASAC Flight Segment and Network Cost Models .....	4
THE ASAC FLIGHT SEGMENT COST MODEL MISSION GENERATOR .....	5
Overview .....	5
Model Inputs .....	8
Model Function .....	9
Model Outputs .....	19
THE ASAC FLIGHT SEGMENT COST MODEL COST TRANSLATOR .....	19
Model Inputs .....	19
Model Operation and Output .....	21
THE ASAC AIR CARRIER NETWORK COST MODEL .....	21
The Network Cost Model as Part of the ASAC Model Chain .....	22
The Network Cost Model as a Stand-Alone Model .....	23
EVALUATING ADVANCED AVIATION TECHNOLOGIES .....	24
Evaluating a New Aircraft Technology .....	24
Evaluating New En Route Air Traffic Management Concept .....	24
Evaluating Airport Capacity Changes .....	25
CONCLUSIONS .....	25
References .....	Ref. 1

---

Appendix A ASAC Flight Segment Cost Model Mission Generator User's Guide (see separate Contents).....	A-1
Appendix B Default Data Provided with the Mission Generator .....	B-1
INPUT DATA FILES—SOURCES AND AVAILABLE OPTIONS .....	B-1
Aircraft Files .....	B-1
Flight Phase Data Files.....	B-3
Load Factors File.....	B-3
RELEASE NOTES .....	B-6
Bad Wind Data Days .....	B-6
The Special Use Area (SUA) Map File.....	B-6
Map File Recommendations.....	B-9
Appendix C The ASAC Flight Segment Cost Model Cost Translator User's Guide .....	C-1
INTRODUCTION .....	C-1
RUNNING THE MODEL .....	C-1
PREPARING THE INPUTS .....	C-3
Network Resource File.....	C-3
Scenario File.....	C-4
Cost Factors File.....	C-5
What is in the Cost Factors File .....	C-5
MODEL EXECUTION.....	C-8
Analyzing the Model Output.....	C-9
Appendix D Flight Segment Cost Model Cost Translator Default Data.....	D-1
DEFINITION OF VARIABLE OPERATING COSTS .....	D-1
SOURCE OF AIRLINE OPERATING COSTS DATA.....	D-2
DEFAULT COST FACTORS .....	D-3
OTHER DEFAULT AND SAMPLE FILES .....	D-4

Appendix E The ASAC Network Cost Model User's Guide .....	E-1
INTRODUCTION .....	E-1
Network Cost Model as Part of the ASAC Model Chain .....	E-1
Network Cost Model—Stand-Alone Mode.....	E-2
RUNNING THE MODEL .....	E-3
PREPARING THE INPUTS .....	E-4
Scenario File.....	E-4
Network Cost File .....	E-6
NETWORK AVERAGE STATISTICS .....	E-6
Flight Profiles File.....	E-7
LOAD FACTORS .....	E-7
Cost Factors.....	E-8
MODEL EXECUTION.....	E-8
Model Execution When Run as Part of the ASAC Model Chain .....	E-8
Model Execution When Run as a Stand-Alone Model .....	E-9
MODEL OUTPUT .....	E-10
Appendix F Network Air Carrier Cost Model Default Data .....	F-1
Appendix G Glossary .....	G-1

## FIGURES

Figure 1. Relationship of ASAC Cost Models.....	5
Figure 2. Flight Segment Cost Model Mission Generator .....	8
Figure 3. Sample Weather Histogram .....	18
Figure 4. Sample Cost Factors File .....	21
Figure C-1. Inputs/Outputs of Model .....	C-2
Figure C-2. Entering the FSCM Cost Translator via the World Wide Web .....	C-2
Figure E-1. Model Inputs and Outputs, Path A .....	E-2

---

Figure E-2. Model Inputs and Outputs, Path B .....	E-3
Figure E-3. Entering the Network Cost Model via the World Wide Web .....	E-4

## TABLES

Table 1. Sample Network Cost Model Output .....	22
Table B-1. Available Aircraft.....	B-1
Table B-2. Available Aircraft (continued) .....	B-2
Table B-3. Available Aircraft (continued) .....	B-3
Table B-4. Flight Phase Data Files.....	B-5
Table D-1. Variable Operating Cost Categories and Elements .....	D-3
Table D-2. Default Cost Factors.....	D-4
Table E-1. Network Cost Model Calculations .....	E-8
Table E-2. Sample Network Cost Model Output .....	E-11
Table F-1. Default Flight Profiles File .....	F-2
Table F-1. Default Flight Profiles File (continued).....	F-3

# Symbol Table

$\hat{a}, b$	Parameters in expression for $dW/dx$
$\rho$	Density, $\text{kg}/\text{m}^3$
$A$	Aircraft reference area, $\text{m}^2$
$a$	Sound speed, $\text{m}/\text{s}$
$C_D$	Drag coefficient
$C_{D0}, C_{D2}, C_{D16}$	Coefficients in BADA model of $C_D$ as function of $C_L$ and $M$ (drag polar coefficients)
$C_L$	Lift coefficient
$D$	Drag $N$
$E$	Energy, Joules
$F$	Fuel flow, $\text{kg}/\text{hr.}$
$f_1, f_2$	Coefficients in BADA fuel flow model
$f_i, f_{i2}$	Coefficients in BADA model of idle thrust
$F_{idle}$	Fuel flow at idle thrust, $\text{kg}/\text{hr.}$
$g$	Acceleration of gravity, $\text{m}/\text{sec}^2$
$h$	Altitude, $m$
$L$	Lift, $N$
$M$	Mach number
OWE	Operating weight empty, $\text{lb}$
$T$	Thrust, $N$

---

$C_{T1}, C_{T2}, C_{T3}$	Coefficients in BADA model of idle thrust
$T_{idle}$	Idle thrust, N
$T_{m1}, T_{m2}, T_{m3}$	Coefficients in BADA model of maximum thrust
$T_{max}$	Maximum thrust, $N$
$V$	Speed, m/s
$V_{kt}$	Speed, knots
$W$	Weight, $N$
$x$	Distance, nmi



# The ASAC Flight Segment and Network Cost Models

## SUMMARY

To meet its objective of assisting the U.S. aviation industry with the technological challenges of the future, National Aeronautics and Space Administration (NASA) must identify research areas that have the greatest potential for improving the operation of the air transportation system. Therefore, NASA seeks to develop the ability to evaluate the potential impact of various advanced technologies. By thoroughly understanding the economic impact of advanced aviation technologies and by evaluating how those new technologies would be used within the integrated aviation system, NASA aims to balance its aeronautical research program and help speed the introduction of high-leverage technologies. To meet these objectives, NASA is building an Aviation System Analysis Capability (ASAC).

NASA envisions ASAC primarily as a process for understanding and evaluating the impact of advanced aviation technologies on the U.S. economy. ASAC consists of a diverse collection of models, data bases, analysts, and other individuals from the public and private sectors brought together to work on issues of common interest to organizations within the aviation community. ASAC also will be a resource available to the aviation community to perform analyses; provide information; and assist scientists, engineers, analysts, and program managers in their daily work.

One of the areas of evaluation is to predict the impact of advanced aviation technologies on air carrier costs. The ASAC Flight Segment Cost Model (FSCM) with its constituent Mission Generator and Cost Translator and the ASAC Air Carrier Network Cost Model address this need.

The FSCM Mission Generator is used to predict aircraft trajectories and associated resources (time and fuel) for one or more individual flights given: aircraft technology; trajectory rule (e.g., "fly the great circle or fly to minimize fuel consumption and arrive on schedule"); and a set of data on winds aloft.

The FSCM Cost Translator converts flight resources into dollar costs. It is designed to work with the FSCM Mission Generator, but it can also be used independently.

The Network Cost Model can be used in one of two ways: It can be used to summarize the costed output from the FSCM of a network of flight segments, or it can be used independently to estimate the costs of flying an airline's network of flights

---

where the number of departures and average stage length of flights flown by each type of aircraft are known.

Each of these models has been designed to work in a highly integrated fashion with the others, as well as to utilize and/or support the other models in the ASAC suite.

The ASAC Flight Segment Cost Model and ASAC Air Carrier Network Cost Model can be used to assess the impact of advanced aviation technologies in a variety of scenarios. To assess the cost impact of a new aircraft technology on airline variable operating costs, an analyst would first run a base case through the ASAC FSCM Mission Generator, ASAC FSCM Cost Translator, and ASAC Air Carrier Network Cost Model. The analyst would then substitute a new hypothetical aircraft utilizing the new technology for one or more aircraft in the base case and repeat the sequence of model runs, comparing the end results. To evaluate the cost impact of a new en route air traffic management concept, the same set of models would be run for a base case and an alternative case using new trajectory generation rules consistent with the new concept. To assess the cost impact of airport capacity improvements, the ASAC FSCM Cost Translator would be used to determine the costs of delays under the base and alternative scenarios as determined by another model such as the ASAC Airport Delay Model. The models are flexible enough to support these and a variety of other analyses.

## INTRODUCTION

### NASA's Role in Promoting Aviation Technology

The United States has long been the world's leader in aviation technology for civil and military aircraft. During the past several decades, U.S. firms have transformed this position of technological leadership into a thriving industry with large domestic and international sales of aircraft and related products.

Despite its historic record of success, the difficult business environment of the recent past has stimulated concerns about whether the U.S. aeronautics industry will maintain its worldwide leadership position. Increased competition, both technological and financial, from European and other non-U.S. aircraft manufacturers has reduced the global market share of U.S. producers of large civil transport aircraft and cut the number of U.S. airframe manufacturers to only two.

NASA's primary role in supporting civil aviation is to develop technologies that improve the overall performance of the integrated air transportation system, making air travel safer and more efficient, while contributing to the economic welfare of the United States. NASA conducts much of the basic and early applied research that creates the advanced technology introduced into the air transportation system. Through its technology research program, NASA aims to maintain and improve the leadership role in aviation technology and air transportation held by the United States for the past half century.

The principal NASA program supporting subsonic transportation is the Advanced Subsonic Technology (AST) program. In cooperation with the Federal Aviation Administration (FAA) and the U.S. aeronautics industry, the goal of the AST program is to develop high-payoff technologies that support the development of a safe, environmentally acceptable, and highly productive global air transportation system. NASA measures the long-term success of its AST program by how well it contributes to an increased market share for U.S. civil aircraft and aircraft component producers and to the increased effectiveness and capacity of the national air transportation system.

### NASA's Research Objective

To meet its objective of assisting the U.S. aviation industry with the technological challenges of the future, NASA must identify research areas that have the greatest potential for improving the operation of the air transportation system. Therefore,

---

NASA seeks to develop the ability to evaluate the potential impact of various advanced technologies. By thoroughly understanding the economic impact of advanced aviation technologies and by evaluating how those new technologies would be used within the integrated aviation system, NASA aims to balance its aeronautical research program and help speed the introduction of high-leverage technologies. To meet these objectives, NASA is building an Aviation System Analysis Capability (ASAC).

## Goal of the ASAC Project: Identifying and Evaluating Promising Technologies

The principal goal of ASAC is to develop credible evaluations of the economic and technological impact of advanced aviation technologies on the integrated aviation system. These evaluations would then be used to assist NASA program managers to select the most beneficial mix of technologies in which NASA should invest, in broad areas, such as propulsion or navigation systems, and in more specific projects within the broader categories. Generally, engineering analyses of this kind require multidisciplinary expertise, possibly using several models of different components and technologies, giving consideration to multiple alternatives and outcomes.

## Technology Impacts on Airline Costs

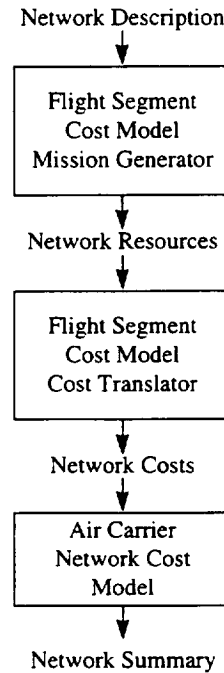
In general, advanced aviation technologies will impact an airline's cost of aircraft operation (direct variable cost) and aircraft ownership (direct capital). Other airline costs such as reservations and facilities are unlikely to be significantly impacted by new aviation technologies. ASAC will include a set of models to evaluate the impact of those new technologies on airline costs, focusing primarily on direct variable costs.

## Overview of the ASAC Flight Segment and Network Cost Models

To estimate the impact of advanced aviation technologies on airline costs, ASAC includes a set of models to estimate the costs of flying individual flight segments—the ASAC Flight Segment Cost Model—and to summarize the costs of flying a network of segments—the ASAC Air Carrier Network Cost Model. The ASAC Flight Segment Cost Model is divided into a Mission Generator and a Cost Translator. The Mission Generator receives a list of flights to be flown and estimates the resources (time and fuel) used on those flights. The Cost Translator converts those resources into financial costs (dollars) on an individual flight basis. The ASAC Air Carrier Network Cost Model summarizes the costs of flying a network of flights. Each of

these models has been designed to work in a highly integrated fashion with the others (see Figure 1), although each can be used independently.

*Figure 1. Relationship of ASAC Cost Models*



These models were developed to support comparative analyses. In order to evaluate the impact of an advanced aviation technology on airline costs, it is necessary to use the appropriate model(s) twice: once on a baseline case and a second time using the alternative technology.

## THE ASAC FLIGHT SEGMENT COST MODEL MISSION GENERATOR

### Overview

#### PURPOSE

The ASAC Flight Segment Cost Model Mission Generator is a tool to determine trajectories flown, fuel consumed, and time by phase of flight for one or more segments.

---

It is designed to be a comparative model—the results produced with one set of inputs is to be compared with the results produced with a similar set of input modified slightly for the advanced aviation technology/policy being analyzed.

## WHAT IS A FLIGHT SEGMENT?

A flight segment is defined for the purpose of this model as

- ◆ a type of equipment,
- ◆ flown between an origin and destination,
- ◆ by an airline,
- ◆ with a scheduled block time determined by scheduled departure and scheduled arrival times,
- ◆ over a period of days.

Model input is in the form of a matrix of flight segments with this and some additional data used by the Cost Translator but most easily generated at the same time as the network of flights.

## ASSUMPTIONS AND THEIR IMPLICATIONS

Two key assumptions drove the development of this model: First, the Mission Generator must support a variety of analyses each with a differing objective and different parameters. Second, the state of the art and sophistication of models and available data to drive models of various phases of flight will change over time. Both of these assumptions dictated the development of a flexible model.

To support a variety of analyses, the model requires little in the way of new input from the users, providing instead a library of default data. However, the model enables the users to substitute new and or modified data in place of the default values.

To be able to accommodate increased knowledge of the aviation system as it relates to either models or data, each phase of flight is addressed in a separate module of the model. Each in turn addresses the time and fuel to conduct the normal operation of a phase of flight and the time and fuel consumed due to delay in that phase of flight.

Either data that drive phases of flight modules or the modules themselves can be easily upgraded as new data or other models become available. For example, one could provide the results of a terminal area productivity model's expected delays by airport and aircraft type to the module that deals with approach delays.

## DESIGN OF THE MODEL

The ASAC FSCM Mission Generator is a model coded in C++ hosted by ASAC on a UNIX server. Remote (and simplified) access is provided by World Wide Web pages available to authorized users at <http://www.asac.lmi.org/>.

A user must provide only two sets of inputs: the list of flight segments to be flown and the rule to use to determine flight trajectories. All other settings, including pointers to default data files, are contained in a scenario file that can be used as provided or modified in whole or in part by the analyst. Details on the operation of the Mission Generator are provided in Appendix A.

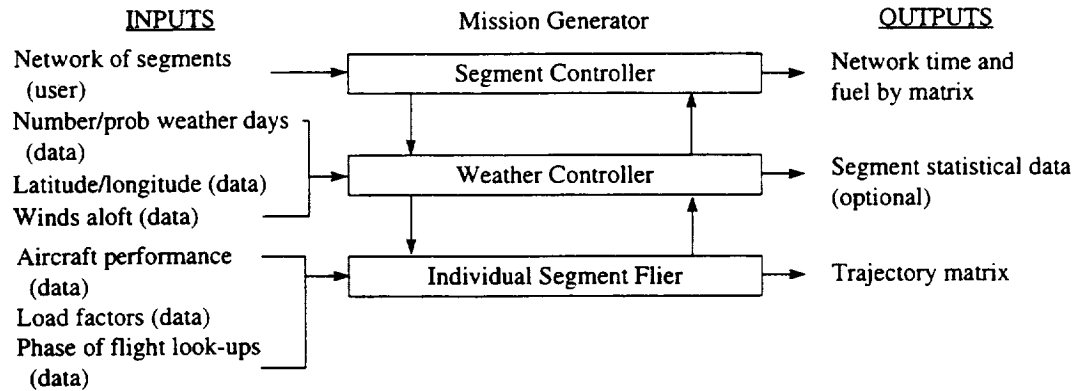
Conceptually, the model consists of three functional tiers of software:

- ◆ *Segment controller* manages individual input segments, each of which can occur for several days. This tier of software enables multiple segments to be contained in one set of input/output files.
- ◆ *Weather controller* manages one segment at a time through the weather conditions (winds aloft) encountered on each day to be flown. This tier of software enables multiple days to be analyzed for each individual segment in the input file.
- ◆ *Segment flier* estimates the time and fuel consumed in each phase of flight for one segment on one day.

Each of these conceptual tiers is actually composed of many program modules.

Figure 2 illustrates these tiers and model input and output.

*Figure 2. Flight Segment Cost Model Mission Generator*



## Model Inputs

The model user must provide the Mission Generator with one or more flight segments to be flown and the trajectory rule for flying those flights. Additionally, a variety of other data are available in a default library. The user may use these data, substitute his/her own files, or modify the default data for use in the analysis being conducted. Detailed contents, formats, and procedures are in Appendix A.

### FLIGHT SEGMENT INPUT—THE NETWORK DESCRIPTION FILE

The user provides the Mission Generator with data on flight segments to be flown via a network description file. This file is a matrix in which each row contains information on the individual flight segment—origin; destination; aircraft; airline; start and end dates (to support multiday evaluations of the effects of winds aloft); scheduled departure time, scheduled arrival time, block time (subtraction of scheduled arrival from scheduled departure adjusted for time zones); and a flights per day factor (used by the ASAC FSCM Cost Translator and Air Carrier Network Cost Model).

A single flight segment network description file may be generated on-line via the World Wide Web interface.

### HOW TO FLY—THE TRAJECTORY RULE

The user provides the Mission Generator with the rule to determine the route to be flown. An example trajectory rule might be “fly the great circle route at a single alti-



tude to arrive on time using minimum fuel” or “fly an optimal path (free flight) with no more than a one-step climb to arrive on time using minimum fuel.” A library of trajectory rules has been provided. Additional rules may be added in a relatively simple manner.

Design of the model with the trajectory rule as an independent input, as opposed to being embedded in the software, gives the model much of its flexibility.

## OTHER DATA—THE SCENARIO FILE

The scenario file provides a simple way to track other model settings as well as substitute new or modify existing default values. It contains the following information:

- ◆ Weather (winds aloft) sampling rule and location of winds aloft data
- ◆ Known aircraft types and the location of aircraft performance data
- ◆ The location of aircraft and airline load factors information
- ◆ The location of phase of flight data for all phases of flight other than cruise (see discussion below)
- ◆ Settings for use in any flight routing optimization
- ◆ Location of output files.

Each of the file and/or directory locations in the Scenario File enables users to substitute completely new files for existing default values. If the users desire to modify existing default data, they may do so via the World Wide Web, which then copies the modified data into a new file for use by the analyst. These files may then be saved for reuse at another time. Details on the generation of model defaults are in Appendix B.

## Model Function

Each of the three functional tiers of software of the Mission Generator are described in the subsections below. We begin with the segment flier (single segment—single day), which is the heart of the model. We then discuss the weather controller (single segment—multiple days) and the segment controller (multiple segments).

---

## FLYING A SINGLE SEGMENT ON A SINGLE DAY—THE SEGMENT FLIER

The segment flier takes advantage of the fact that each phase of flight is mostly independent, with the exception of the cruise phase when an aircraft has the ability to alter its trajectory or speed (within limits) to account for time spent in other phases. For purpose of this model, the phases of flight are

- ◆ actual departure time,
- ◆ taxi out,
- ◆ take off,
- ◆ climb,
- ◆ cruise,
- ◆ approach and landing, and
- ◆ taxi in.

For each phase of flight, the model estimates resources consumed both to accomplish the actual operation and during any delays encountered during the phase. With the exception of the cruise operation phase, the Mission Generator currently treats the time spent in each phase of flight as a data look-up and calculates fuel consumed and/or impact on the cruise operation phase. Appendix B discusses default look-up data generation for phases other than cruise operation. Each phase is discussed in a subsection below.

### Actual Departure Time Phase

This phase has no operation time, only a delay time on the basis of the airport of origin. Default data are based on Airline Service Quality Performance (ASQP)-reported delays.

### Taxi-out Phase

Time spent in taxi-out operations and delays is based on origin airport and airline (gate location). Default data for this phase is based on FAA-reported values for the main runway. Fuel use is calculated at idle throttle for the time spent on the ground. We have not adjusted for surface weather conditions, although doing so would only require the analyst to compute weighted average operation and delay.

### Takeoff Phase

Time spent during takeoff operations was calculated once for each aircraft type; that information serves as default look-up data. No delays occur during take off normally.

### Climb Phase

Climb is defined as the interval from takeoff to a given altitude is dependent on the aircraft. Time and distance traveled were computed once for each aircraft for a maximum rate of climb; that information serves as default look-up data. No delay times were calculated because delays impact operations at cruise, not during climb.

### Cruise Phase

The model computes the cruise phase after all other phases since most rational trajectory rules are based on scheduled arrival time. Similarly, cruise delays are first looked up before the actual conduct of cruise operations is modeled. Cruise delays are assumed to occur as an order to the flight to slow down rather than as a rerouting. As such, they impact the block time objective of the cruise phase rather than consuming additional fuel during the delay. In many cases, this actually will save the flight fuel, though causing it to arrive somewhat later. In reality, vectoring in specific FAA air traffic management sectors does sometimes occur, with flights able to make up time elsewhere in their trajectories. As additional data and/or models of en route delays become available, the Mission Generator could accommodate them as either data in look-up tables or by modifying the modules in this phase of flight. Currently, data on en route delays are not available.

### *Cruise Operations Phase*

This phase of flight is the only one modeled in detail in the Mission Generator. The three independent components used in this phase are

- ◆ an aircraft physical model, which uses an aircraft performance model to calculate fuel, time, and distance traveled for parameters passed from the optimizer;
- ◆ optimization software that varies trajectory parameters; and
- ◆ a project controller that converts the trajectory rule into optimization settings and handles the interface between the optimizer and the aircraft physical model.

---

We used Eurocontrol's BADA models [1] and data as our principal aircraft performance models for this study. Our aircraft physical model is described below. [1] We compared the results using this modeling suite with that from both Aircraft Synthesis (ACSYNT) and Flight Optimization System (FLOPS) and found them to be comparable for selected aircraft. BADA was selected for ease of computation and because it is the only readily available model with an extensive library of aircraft data. Appendix A provides instructions on how to convert either ACSYNT or FLOPS output into the equivalent BADA data.

The following paragraphs briefly review the specific aircraft physical model that we used to compute flight times and fuel consumed. Reference [2] gives a more complete discussion.

The basic aircraft performance models required are drag as a function of lift, Mach number, and altitude, and fuel flow as a function of thrust, Mach number, and altitude. The BADA models provide these in this way:

Lift coefficient,  $C_L$ , is computed as

$$C_L = \frac{2L}{\rho V^2 A}. \quad [\text{Eq. 1}]$$

(Note: All symbols are defined in the symbol table, page vi.)

The BADA models provide values for reference area  $A$  for several aircraft. The product of lift and the ratio of drag coefficient,  $C_D$ , to lift coefficient,  $C_L$ , gives drag,  $D$ . In the BADA models,

$$C_D = (C_{D0} + C_{D2} C_L^2)(1 + C_{D16} M^{16}), \quad [\text{Eq. 2}]$$

where  $C_{D0}$ ,  $C_{D2}$ , and  $C_{D16}$  are aircraft-specific parameters given in the BADA data. Thus, in the BADA models,

$$D = L \left( C_{D0} \frac{\rho M^2 a^2 A}{2L} + C_{D2} \frac{2L}{\rho M^2 a^2 A} \right) (1 + C_{D16} M^{16}). \quad [\text{Eq. 3}]$$

In Equation 3, we have replaced speed  $V$  with the product of Mach number  $M$  and sound speed  $a$ . Since density,  $\rho$ , and sound speed,  $a$ , are functions of altitude (we used the 1976 U.S. standard atmosphere for these functions), Equation 3 gives the required expression for drag as a function of lift, altitude, and Mach number.

The BADA models give alternate expressions for fuel flow as a function of thrust, altitude, and Mach number for turbojet and turboprop airplanes. For turbojets, fuel flow,  $F$ , is

$$F = Tf_1 \left( 1 + \frac{Ma}{f_2} \right) \text{kg/hr.}, \quad [\text{Eq. 4}]$$

where  $f_1$  and  $f_2$  are aircraft-specific constants from the BADA library.<sup>1</sup> For turbo-props,

$$F = TMaf_{T1} \left( 1 - \frac{Ma}{f_{T2}} \right) \text{kg/hr.} \quad [\text{Eq. 5}]$$

In addition to the basic aircraft performance Equations 3, 4, and 5, we used several other BADA models in computing complete optimal trajectories. Specifically, for idle thrust we used

$$T_{idle} = C_{T1} \left( 1 - \frac{h}{C_{T2}} + C_{T3}M \right) N, \quad [\text{Eq. 6}]$$

where  $C_{T1}$ ,  $C_{T2}$ , and  $C_{T3}$  are given for each aircraft in the BADA library.

For fuel flow at idle thrust, we used

$$F_{idle} = f_i \left( 1 - \frac{h}{f_{i2}} \right) \text{kg/hr.} \quad [\text{Eq. 7}]$$

Finally, we used the BADA models for maximum thrust, which are

$$T_{max} = T_{M1} \left( 1 - \frac{h}{T_{M2}} + \frac{T_{M3}}{h^2} \right) N \quad [\text{Eq. 8}]$$

for turbojets and

---

<sup>1</sup> In some cases, our output units differ from the units of the BADA library.

---


$$T_{\max} = \frac{T_{M1}}{Ma} \left( 1 - \frac{h}{T_{M2}} \right) + T_{M3} \text{ N} \quad [\text{Eq. 9}]$$

for turboprops.

The BADA models enable one to write an exact expression for fuel burned during flight of a specified distance at constant altitude and Mach number. This result, which we found convenient in optimal route calculations, may be developed as follows: For flight at constant speed and altitude, the time rate of change of aircraft weight,  $W$ , is given by

$$\frac{dW}{dt} = -gF \frac{N}{hr},$$

and the rate of change of  $W$  with respect to distance flown,  $x$ , is

$$\frac{dW}{dx} = -\frac{gF}{V_{kt}} \frac{N}{nmi},$$

where  $V_{kt}$  denotes speed in knots. For flight at constant altitude and speed,  $L = W$  and  $T = D$ . Using these facts and Equations 3 and 4 leads to an expression for  $\frac{dW}{dx}$  of the form

$$\frac{dW}{dx} = -\hat{a} - bW^2.$$

By elementary means, one finds the solution of this equation that takes the value  $W_0$  when  $x = x_0$  is

$$W = \sqrt{\frac{\hat{a}}{b}} \tan \left[ \tan^{-1} \left( \sqrt{\frac{b}{\hat{a}}} W_0 \right) - \sqrt{\hat{a}b} (x - x_0) \right] \text{ N},$$

where the terms involving parameters  $a$  and  $b$  may be written

$$\sqrt{\frac{\hat{a}}{b}} = \frac{pv^2}{2} A \sqrt{\frac{C_{D0}}{C_{D2}}} \text{ N}$$

and

$$\sqrt{\hat{a}b} = \frac{gf_1}{v_{kr}} \left(1 + C_{D16} M^{16}\right) \left(1 + \frac{v}{f_2}\right) \sqrt{C_{D0} C_{D2}}.$$

We modeled fuel burns during cruise segments using the exact solution and adjusted for any altitude and speed changes by the total energy method in this way. During flight at constant altitude and speed, fuel flows at the rate,  $F$ , given by Equation 4; the propulsion system delivers energy at the rate,  $Tv$ . If thrust were increased by an amount,  $\delta T$ , for a time,  $\delta t$ , with speed constant, the propulsion system would add energy  $\delta E = \delta T v \delta t$ . This would add a total fuel burn of  $F \delta T \delta t / T$ . Thus, the fuel burn required to add energy,  $\delta E$ , is  $\delta E \frac{F}{vT}$ . If altitude and velocity changed during a segment, we computed the net energy change, and the net fuel burn required to produce it, with this model, using values of weight, speed, and altitude at the segment's midpoint. We allowed altitude and speed decreases to reduce fuel burn, but never to less than the burn that would have been incurred if the airplane operated at flight idle for the entire segment. Also, we enforce the maximum thrust constraint of the BADA models.

In computing optimal trajectories, we used an analysis module that performs path optimization with gradient-based search techniques. We employed several existing software technologies to develop the modeling capability quickly, and we focused on improving the integrity of the aircraft performance model. Rapid model integration was realized through the Dynamic Integration System, [3] [4] a unique integration architecture that connects several C++ objects in the model. Also, we used the Optimization Workbench [5] to bring minimization capability to the problem immediately. Finally, we employed configuration management features of the Optimization Workbench to manage multiple optimization problems.

The Optimization Workbench's minimization engine is the Design Optimization Tools routine, which is distributed by Phoenix Integration for Vanderplaats Research and Development, Inc., Colorado Springs, CO. Design Optimization Tools makes several optimization methods available. For constrained optimization, the user can select from three methods: Modified Method of Feasible Directions (MMFD), Sequential Linear Programming (SLP), or Sequential Quadratic Programming (SQP). For unconstrained optimization, two methods are available: the Broydon-Fletcher-Goldfarb-Shanno variable metric method or the Fletcher-Reeves conjugate gradient method.

We used the Dynamic Integration System within a trajectory manager to provide a consistent interface to the fuel-burn model. This allows the Optimization Workbench

---

to interact with the trajectory manager indirectly through dynamic variables. Fuel burns and route times are computed on the basis of route and flight conditions.

The Optimization Workbench provides a useful set of tools to manage and coordinate multiple problem formulations. The user can store multiple optimization formulations or configurations within one project. The underlying software is constructed to handle multiple problems and their results efficiently.

Individual configurations are managed under a control object called a “project.” A “configuration” is used here as a specific problem formulation for submission to the optimization algorithm. A configuration consists of a selected set of design variables, constraints, and an objective function. The Optimization Workbench gives the capability to add, delete, or edit configurations within a project. We used this configuration management approach to construct the various trajectory rules.

#### Approach and Landing Phase

We calculated the time spent in descent once for each aircraft type at idle thrust and maximum lift-to-drag ratio; that information serves as default data for approach and landing operations. Approach delays are based on destination city and are assumed to occur at 10,000 feet. The 10,000-foot altitude is adjustable, but delay fuel consumption rates and descent time and fuel-burn look-up tables would need to be adjusted as well. No quality data are available on approach delays since only performance versus schedule is reported by the airlines. Since schedules already contain some buffer for expected delays and do not identify in which phase of flight delays occur, we were unable to use ASQP data for this purpose. The FAA does not track delay data in a usable fashion for our purpose (counts, not minutes, are tracked; aircraft that are vectored as opposed to put into holding patterns are not reported in the counts).

#### Taxi-in Phase

Time spent in taxi-in operations and delays is based on the destination airport and airline (gate location). Default data for this phase is based on FAA-reported values for the main runway. Fuel is calculated at idle throttle for the time spent on the ground.

#### FLYING IN A VARIETY OF WIND CONDITIONS—THE WEATHER CONTROLLER

We have just described how the Mission Generator flies one aircraft from one city to another, given one day’s data on winds aloft. Since the winds aloft change from one day to the next, we designed the weather controller to fly the same aircraft between the same two cities in a variety of wind conditions.



NASA Goddard Space Flight Center's Atmospheric Chemistry and Dynamics Branch told us that 1995 was as typical a year as any in terms of atmospheric winds. Data on eastward and northward components of the winds aloft in 1995 were obtained from the National Oceanic and Atmospheric Administration and serve as default data for the Mission Generator. Four days' data either were missing or incorrect; we have repeated the prior days' data for those after checking that day-to-day variation did not seem overly large.

In order to evaluate an advanced aviation technology's impact using the Mission Generator, one must either fly for many days or a representative sample of days. One of the scenario file parameters is a sampling mode. If sampling mode is off, a flight segment is flown for each day of the period. If sampling mode is on, the user must provide a weather histogram file, which tells the model which sample days to use. In either case the weather controller computes average (or weighted average) and variances of flight times and fuel burns. If users choose to, individual day trajectories can also be saved.

If an analyst chooses to use sample days to represent the weather, he or she must provide a weather histogram file. In this file, the analyst divides the year into periods (for instance seasons or months) with start and end dates. For each period, sample days and their probability of occurrence are then listed. This distribution of representative days will then be used for any day in the period. A sample weather histogram might look like Figure 3.

We performed no analysis to identify sample weather days. No default weather histogram with inherent assumptions on future weather patterns is provided.

*Figure 3. Sample Weather Histogram*

```
[Period_Information]
Number_Of_Periods = 12

Period 1 = JAN; Period 2 = FEB; Period 3 = MAR; Period 4 = APR;
Period 5 = MAY;
Period 6 = JUN; Period 7 = JUL; Period 8 = AUG; Period 9 = SEP;
Period 10 = OCT;
Period 11 = NOV; Period 12 = DEC

[JAN]
Period_Length = 000 31
Day = 001 .2
Day = 009 .2
Day = 015 .2
Day = 024 .2
Day = 028 .2

[FEB]
Period_Length = 032 059
Day = 032 1.0

[MAR]
Period_Length = 060 090
Day = 060 1.0

[APR]
Period_Length = 091 120
Day = 091 1.0

[MAY]
Period_Length = 121 151
Day = 121 1.0

[JUN]
Period_Length = 152 181
Day = 152 1.0

[JUL]
Period_Length = 182 212
Day = 182 1.0

[AUG]
Period_Length = 213 243
Day = 213 1.0

[SEP]
Period_Length = 244 273
Day = 244 1.0

[OCT]
Period_Length = 274 304
Day = 274 1.0

[NOV]
Period_Length = 305 334
Day = 305 1.0

[DEC]
Period_Length = 335 365
Day = 335 1.0

Notes: In this example, January (days 001-031) is represented by 5 days, with each day
equally weighted; but only 1 day would represent each of the other months. Sample is shown in
two columns, actual file would be sequential.
```

## FLYING MULTIPLE SEGMENTS AUTOMATICALLY—THE SEGMENT CONTROLLER

The segment controller manages user inputs and model outputs as well as forwarding a row (one flight segment) of network description data to the weather controller. In this way, an analyst can run a complete or representative network of segments at once. Web-based access is limited to networks that will run in less than a day, or about 40 city pairs for a full year. Longer runs can be accommodated by contacting the ASAC system administrator.

## Model Outputs

Normal model output consists of a network resource file that echoes the scenario file in its header and adds additional fields on flight resources (fuel and time) by phase of flight, variances (fuel and time) by total flight, and other statistics such as flight distance relevant to the flight and needed by the Cost Translator or Network Cost Model.

Individual flight trajectories can be saved as an option, but these files can get quite large and unwieldy. Trajectories are not necessary for many types of analyses.

## THE ASAC FLIGHT SEGMENT COST MODEL COST TRANSLATOR

The ASAC Flight Segment Cost Model Cost Translator is a tool to convert flight resources into dollar costs. It receives a network of flight resource inputs from either the ASAC FSCM Mission Generator or a user-provided table of airline and aircraft combinations. It combines these with a cost factors table to estimate dollar costs by cost category (e.g., flight personnel). Appendix C is the ASAC FSCM Cost Translator user's guide.

## Model Inputs

### FLIGHT RESOURCES

The model operates in one of two modes—(1) either as part of a chain of models such that its main input comes directly from the ASAC FSCM Mission Generator in the form of a network resource file with full resource inputs (fuel and time) for individual flight segments or (2) from user input of airline and aircraft and partial resource in-

---

puts (for instance minutes of delay on the ground).<sup>2</sup> The model is flexible enough to handle these very different cases.

## COST FACTORS

The model is flexible because it relies on a cost factors table to tell it the costs to calculate, the drivers of those costs, and the equations to use in calculations.

- ◆ *Cost drivers* are identified first; departures, time, and fuel can drive costs.
- ◆ *Cost factors* are then identified; these are independent costs driven by one or more of the drivers. Normal cost factors include flight personnel, fuel, maintenance, and other variable costs, although users may define any cost factors they choose.
- ◆ *Cost categories* are identified next; these are costs that are dependent on cost factors. For example, variable operating cost could be defined as the sum of flight personnel, fuel, maintenance, and other variable costs. The users may also define any cost categories they choose.
- ◆ *Cost data* are then defined for each airline and aircraft combination. Data in each row consist of the airline, aircraft, and a constant cost and variable cost for each cost driver for each cost factor. Thus, if there were three cost drivers and four cost factors, each data row would contain 16 data elements (a constant and three driver coefficients for each cost factor).

A sample cost factors table for input from the Mission Generator (cost of a single flight) is shown in Figure 4.

A cost factors table for use with other models might have fuel cost as a function of time since estimates of actual fuel consumption would not be available.

Default cost factors tables have been provided for use with the model. See Appendix D for details.

---

<sup>2</sup> Users have to provide appropriate cost factors for use when time and fuel for entire flights are not provided.

Figure 4. Sample Cost Factors File

```

NUM_DRIVERS=2
//All costs are driven by two factors defined below.
NUM_FACTORS=4
//Number of detailed costs defined below.
NUM_CATEGORIES=1
//Number of ways to roll up costs. Each defined below.
DRIVERS=T F
//The two cost drivers are T (time-hours) and F (fuel-pounds)
FACTORS=LABOR FUEL MAINT OTHER
//The four cost factors are LABOR, FUEL, MAINT, and OTHER
//A matrix of formulas for various combinations of airlines and equipment fol-
low in the table below.
//For each there is a constant and a coefficient for each driver.
//In this case 4 factors times three elements (constant and two coefficients)
//or twelve data elements for each airline/equipment combination
CATEGORIES=VOC
//The name of the one cpst category (rollup), Variable Operating Cost
VOC=LABOR FUEL MAINT OTHER
VOC_CONST=0
//These two lines define the formula for VOC
//VOC=LABOR+FUEL+MAINT+OTHER+0

```

//Airline	Equip	Labor\$ constant	Labor\$/ Hour	Labor\$/ #fuel	Fuel\$ constant	Fuel \$/ Hour	Fuel\$/ #Fuel	Maint.\$ constant	Maint \$/Hour	Maint\$/ #Fuel	Other Cost\$ Constant	Other Cost\$/ Hour	Other Cost\$/ #Fuel
CO	747	0	1465	0	0	0	0.12	0	1687.06	0	0	7.52	0
NW	747	0	1986.3	0	0	0	0.12	0	1519.95	0	0	20.28	0
TW	747	0	1720.1	0	0	0	0.12	0	1621.36	0	0	22.77	0
UA	747	0	2179.1	0	0	0	0.12	0	1323.00	0	0	9.12	0

**Note:** In this table, labor, maintenance, and other costs are driven by hours and fuel is a constant 12 cents/#.

## Model Operation and Output

The operation of the model is fairly straightforward. The costs of each row of input data are calculated for each of the cost factors and cost categories in the cost factors table. Columns for each of these are appended to the end of the input resource file. If the model is being run in stand-alone mode, as opposed to providing data to the ASAC Air Carrier Network Cost Model, totals are also calculated.

## THE ASAC AIR CARRIER NETWORK COST MODEL

The ASAC Air Carrier Network Cost Model is a simple tool designed to enable an analyst to estimate the costs of flying a network of flights. It operates in one of two

modes—as part of a model chain where input is received from the ASAC FSCM Cost Translator, or in stand-alone mode where network summary data are provided by the user.

In each case, the model summarizes costs by user-defined category and pertinent network summary statistics by airline and aircraft type. Details on the model's use are contained in Appendix E.

## The Network Cost Model as Part of the ASAC Model Chain

When operating as part of the ASAC model chain, the Network Cost Model receives detailed cost data from the ASAC FSCM Cost Translator for each flight segment. The model calculates the number of departures, miles flown, available seat miles, revenue passenger miles, and cost totals by factor and category. It sums these for each airline and aircraft type and produces an output table as shown in Table 1.

*Table 1. Sample Network Cost Model Output*

Airline	Aircraft	Departures	Block Time	Fuel (#)	Miles Flown	ASM	RPM
AA	747	100000	51600	9867000	100000000	10000000000	7000000000
AA	757	80000	51887120	347584000	120000000	24000000000	16080000000
AA	767	65000	40808040	279579950	84500000	33800000000	16900000000
AA	TOT	245000	92746760	637030950	304500000	67800000000	39980000000

Note: ASM = Available Seat Miles, RPM = Revenue Passenger Miles

Airline	Aircraft	Labor	Fuel	Maintenance	Other Cost	Variable Operating Costs
AA	747	\$46,509,656	\$1,184,040	\$21,168,900	\$689,376	\$69,551,971
AA	757	\$50,852,490,000	\$41,710,078	\$18,425,116,250	\$569,720,547	\$69,889,035,000
AA	767	\$41,538,095,625	\$33,549,593	\$24,978,600,156	\$232,605,817	\$66,782,848,438
AA	TOT	\$92,437,095,281	\$76,443,711	\$43,424,885,306	\$803,015,739	\$136,741,435,409

In calculating available seat miles and revenue passenger miles, the Network Cost Model uses the same load factors table that was used by the Mission Generator, the location of which is contained in the Scenario File header in the input network cost file.

## The Network Cost Model as a Stand-Alone Model

In the stand-alone mode, the user generates a network average statistics file summarizing operations. Additionally, the model uses a standard flight profile table, a cost factors table, and a load factors file to calculate flight resources. We provide default files for each of these, although the user may provide new files or modify the existing defaults (default data for this model are discussed in Appendix F). The function of these files is described below.

- ◆ The network average statistics file (table) contains in each row data about airline, aircraft, number of departures, and average stage length. It is used to define the network.
- ◆ The standard flight profile table for each equipment type summarizes the time and fuel consumed and the distance traveled in all noncruise phases of flight; it provides a variable time and fuel consumed, as a function of cruise distance. This table is used to simulate (much less accurately) the function of the ASAC FSCM Mission Generator.
- ◆ The cost factors table used by the Network Cost Model is identical to that used by the ASAC FSCM Cost Translator. It is used to simulate the function of the Cost Translator.
- ◆ The load factors file is used (as described earlier) to calculate available seat miles and revenue passenger miles flown.

The Network Cost Model is much less accurate when used in this mode as instead of as part of the Flight Segment Cost Model to-Network Cost Model chain. However, if the analyst wants to do “what if” analysis on cost factors or flight profiles, this is a valuable tool for first approximation.

---

## EVALUATING ADVANCED AVIATION TECHNOLOGIES

The ASAC Flight Segment Cost Model and ASAC Air Carrier Network Cost Model could be used to assess the impact of advanced aviation technologies in a variety of scenarios, such as

- ◆ new aircraft technology,
- ◆ new en route air traffic management concept, and
- ◆ airport capacity increases.

### Evaluating a New Aircraft Technology

To assess the cost impact of a new aircraft technology, an analyst would employ both the ASAC Flight Segment Cost Model and the ASAC Air Carrier Network Cost Model.

First, a base case would be run through the complete model chain:

*FSCM Mission Generator → FSCM Cost Translator → Network Cost Model*

The analyst then would substitute a new hypothetical aircraft utilizing the advanced technology for one or more aircraft in the base-case scenario and repeat the sequence of model runs.

### Evaluating New En Route Air Traffic Management Concept

To assess the cost impact of a new en route air traffic management concept, an analyst would employ both the ASAC Flight Segment Cost Model and the ASAC Air Carrier Network Cost Model.

First, a base case would be run through the complete model chain:

*FSCM Mission Generator → FSCM Cost Translator → Network Cost Model*

Then, the same network of flight segments would run through the same model chain using the alternative air traffic management concept to create a new trajectory rule to be used by the Mission Generator.



## Evaluating Airport Capacity Changes

To assess the cost impact (savings) of airport capacity improvements, the ASAC FSCM Cost Translator would be used to determine the costs of delays under the base and alternative scenarios as determined by another model such as the ASAC Airport Delay Model. These costs could then be compared to assess the benefits of the various options.

## CONCLUSIONS

The set of models described in this report provides a powerful toolbox for analyzing the cost impacts of advanced aviation technologies on air carrier operations.

The ASAC FSCM Mission Generator is used to predict aircraft trajectories and associated resources (time and fuel) for one or more individual flights given aircraft technology, trajectory rule (e.g., fly great circle or fly to minimize fuel consumption and arrive on schedule), and a set of data on winds aloft.

The FSCM Cost Translator converts flight resources into dollar costs. It is designed to work with the FSCM Mission Generator, but it can also be used independently.

The ASAC Air Carrier Network Cost Model can be used in one of two ways: It can be used to summarize the costed output from the FSCM of a network of flight segments, or it can be used independently to estimate the costs of flying an airline's network of flights where the number of departures and average stage length of flights flown by each type of aircraft are known.

These models were designed to work in either an integrated fashion or as stand-alone tools, depending on the analysis at hand.

The ASAC FSCM Mission Generator is the most detailed of the models, notably in treatment of the cruise phase of flight. Because of its flexible design, it can easily accommodate new data or routines that model other phases of flight.



# References

- [1] European Organization for the Safety of Air Navigation, Eurocontrol Experimental Center, *User Manual for the Base of Aircraft Data (BADA), Revision 2.4f*. European Economic Community Note, May 1996, Bretigny France.
- [2] D. Lee, et al., *Technical and Economic Analysis of Air Transportation Management Issues Related to Free Flight*, LMI Report NS501T1, McLean, VA, January 1997.
- [3] S.A. Woyak and A. Myklebust, "Functionality and Data Integration of Software Modules through Dynamic Integration," accepted for publication in *Integrated Computer-Aided Engineering*.
- [4] S.A. Woyak, A. Myklebust, and B. Malone, "An Architecture for Creating Engineering Applications: The Dynamic Integration System," 15<sup>th</sup> Annual International Computers in Engineering Conference, Boston, MA, September 17-20, 1995.
- [5] B. Malone and S.A. Woyak, "An Object-Oriented Analysis and Optimization Control Environment for the Conceptual Design of Aircraft," AIAA Paper 95-3862, First AIAA Aircraft Engineering, Technology, and Operation Congress, Los Angeles, CA, September 1995.



Appendix A  
ASAC Flight Segment Cost Model Mission Generator  
User's Guide

---



# Contents

---

## Appendix A ASAC Flight Segment Cost Model Mission Generator

User's Guide.....	A-1
INTRODUCTION .....	A-9
THE FLIGHT SEGMENT.....	A-10
RUNNING THE MODEL .....	A-12
PREPARING MODEL INPUTS .....	A-13
The Network Description .....	A-13
The Trajectory Rule.....	A-16
Other Inputs: The Scenario File .....	A-19
MODEL OPERATION.....	A-21
Single Flight Modeling.....	A-21
Vehicle Modeling .....	A-22
Initial and Final Waypoint Cruise Altitudes.....	A-29
SINGLE FLIGHT IN A VARIETY OF WINDS—WEATHER SAMPLING .....	A-33
Single Flight Modeling.....	A-33
Multiple Flight Modeling .....	A-37
MODEL OUTPUT .....	A-37
The Network Resource File.....	A-37
Other Optional Output.....	A-38
SAMPLE RESULTS .....	A-40

## Annex A-1 ASAC Flight Segment Cost Model Mission Generator File

Formats.....	A-45
NETWORK DESCRIPTION FILE.....	A-45
Purpose .....	A-45
Description .....	A-45

---

Leap Year .....	A-46
SCENARIO FILE.....	A-47
Purpose .....	A-47
Description .....	A-47
Explanation of File Tags .....	A-48
NETWORK RESOURCE FILE.....	A-51
Purpose .....	A-51
Description .....	A-51
AUXILIARY FILES .....	A-53
Aircraft Name List (Index) File.....	A-53
Aircraft Data File.....	A-55
Weather Histogram File .....	A-57
Trajectory Phase Lookup Files .....	A-61
Aircraft Load Factors File .....	A-63
User-Specified Routing Input Data File .....	A-64
Annex A-2 Constructing a New Trajectory Rule.....	A-67
GENERAL FORMAT FOR A TRAJECTORY RULE .....	A-67
What Is a Design Variable in the Project File? .....	A-69
Annex A-3 Route Database Utility .....	A-73
PROGRAM OPERATION.....	A-74
USER-SPECIFIED ROUTING INPUT DATA FILE .....	A-74
Annex A-4 Converting ACSYNT Output to Mission Generator Aircraft Format .....	A-77
CREATING AN AIRCRAFT DATA FILE USING ACSYNT INFORMATION .....	A-78
PERFORMANCE AND VEHICLE LIMITS .....	A-78
WEIGHTS .....	A-83
GEOMETRIC PARAMETERS.....	A-86
MISCELLANEOUS.....	A-86



AERODYNAMICS .....	A-87
PROPULSION: FUEL FLOWS .....	A-89
Steps for creating a fuel flow model from ACSYNT output. ....	A-92
PROPULSION: THRUST MODEL .....	A-94
Steps for Creating a Thrust Model From ACSYNT Output. ....	A-95
Obtaining Fuel-Burn Rates from ACSYNT Output (Trajectory).....	A-98
Converting FLOPS Output to Mission Generator Aircraft Format .....	A-99
PERFORMANCE AND VEHICLE LIMITS .....	A-99
WEIGHTS .....	A-103
GEOMETRIC PARAMETERS .....	A-105
MISCELLANEOUS .....	A-106
AERODYNAMICS .....	A-106
PROPULSION: FUEL FLOWS .....	A-109
Steps for Creating a Fuel Flow Model From FLOPS Output.....	A-113
PROPULSION: THRUST MODEL .....	A-115
Steps for Creating a Thrust Model From FLOPS Output. ....	A-115
Obtaining Fuel-Burn Rates from FLOPS Output (Trajectory).....	A-118

## FIGURES

Figure A-1. Flight Planning Route Options .....	A-10
Figure A-2. The Phases of a Flight Segment.....	A-11
Figure A-3. Entering the ASAC FSCM Mission Generator via the World Wide Web .....	A-12
Figure A-4. NDF Entry .....	A-13
Figure A-5. Scenario File Entry Screen.....	A-20
Figure A-6. Building a Scenario File on the Scenario File Customizer Options Screen ...	A-21
Figure A-7. Flight Path Segment between Two Cities Showing Individual Waypoints .....	A-28
Figure A-8. Segment Description Used Between Two Waypoints .....	A-29

---

Figure A-9. Example of Wind Stream Lines at 38,000 Feet on January 1, 1995.....	A-31
Figure A-10. Average per-Flight Cruise Phase Fuel Usage for September for 13 Flight Segments.....	A-40
Figure A-11. Altitude Profiles for Days 1 to 20 for a Sample Flight Segment with Mach/Altitude Optimization .....	A-41
Figure A-12. Comparison of Waypoints for Two Trajectory Rules.....	A-42
Figure A-13. Comparison of Great Circle and Optimized Routes, Shown with a Wind Field.....	A-43
Figure A-14. Comparison of Altitudes for Two Trajectory Rules .....	A-44
Figure A-2-1. Physical Description of Design Variables .....	A-70
Figure A-4-1. Trajectory Output .....	A-83
Figure A-4-2. Weight Statement Example .....	A-84
Figure A-4-3. Summary—ACSYNT Output .....	A-86
Figure A-4-4. Aerodynamic Output from ACSYNT .....	A-88
Figure A-4-5. Aero Graph File .....	A-89
Figure A-4-6. Propulsion Output Section .....	A-91
Figure A-4-7. Trajectory Output .....	A-94
Figure 8. Thrust Curve Model Constructed from ACSYNT Data .....	A-97
Figure A-4-9. Trajectory Output .....	A-98
Figure A-4-10. Weight Statement .....	A-104
Figure A-4-11. Aerodynamic Data.....	A-106
Figure A-4-12. Aerodynamic Output From FLOPS.....	A-107
Figure A-4-13. Propulsion Output Section .....	A-111
Figure A-4-14. Linear Model Derived from FLOPS.....	A-114
Figure A-4-15. Thrust Curve Model Constructed from FLOPS Data.....	A-118
Figure A-4-16. Mission Summary.....	A-119

## TABLES

Table A-1. Far North Airlines Fleet .....	A-14
---	------

Table A-2. Far North Airlines Flight Operations .....	A-14
Table A-3. Sample NDF for Far North Airlines .....	A-14
Table A-4. Trajectory Rules Available .....	A-18
Table A-5. FAA-Approved Altitudes.....	A-19
Table A-6. Winds Aloft Data Points .....	A-30
Table A-1-1. Record Line Descriptions .....	A-46
Table A-1-2. Abbreviations for Line Results.....	A-52
Table A-1-3. Additional Abbreviations for Line Results.....	A-53
Table A-1-4. Lookup Keys for Trajectory Files.....	A-62
Table A-2-1. Available Design Variable Parameters for Trajectory Rule Creation.....	A-70
Table A-2-2. Available Objective Functions Variables for Trajectory Rules.....	A-71
Table A-4-1. Recognizing Maximum Altitude .....	A-80
Table A-4-2. Sample Computation of Velocity Correction Coefficient Cf2 .....	A-93
Table A-4-3. Sample Thrust Coefficients .....	A-97
Table A-4-4. Summary Data For Cruise Schedule 1.....	A-103
Table A-4-5. Propulsion Format .....	A-111
Table A-4-6. Sample Computation of Velocity Correction Coefficient Cf2 .....	A-114
Table A-4-7. Sample Computation of Velocity Correction Coefficient Cf2 .....	A-117



## INTRODUCTION

The ASAC Flight Segment Cost Model (FSCM) Mission Generator is a tool to determine trajectories flown, fuel consumed, and block time by phase of flight for one or more segments. It is designed to be a comparative model—the results produced with one set of inputs is to be compared with the results produced with a similar set of input modified slightly for the advanced aviation technology/policy being analyzed.

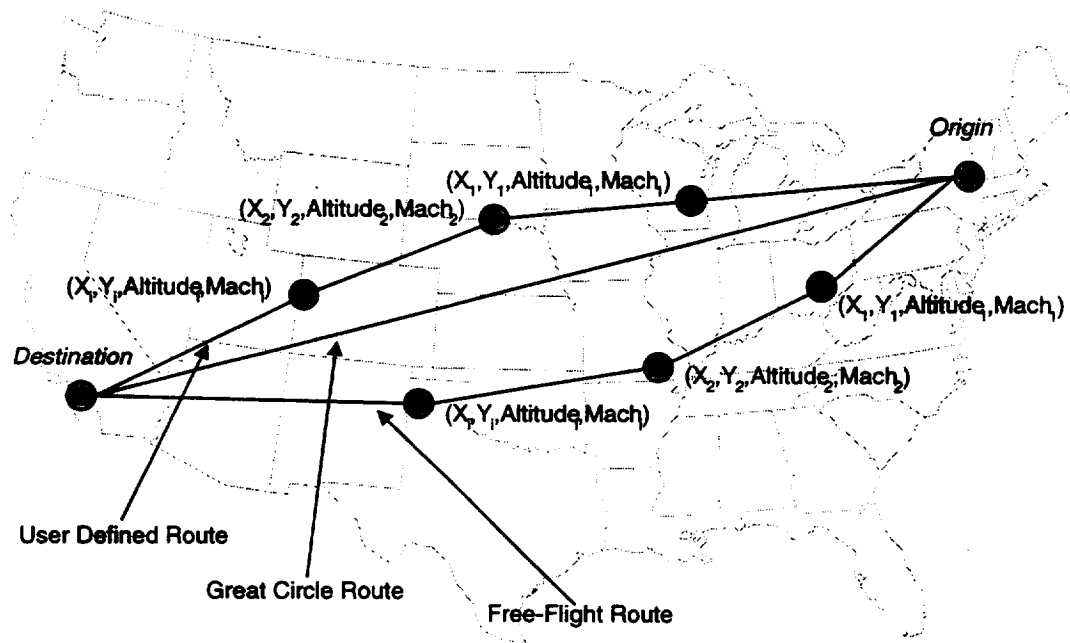
The Mission Generator program estimates elapsed time and consumed fuel, along with other data, for flights under various winds aloft. At the heart of the Mission Generator lies a flight optimizer capable of determining flight trajectories and resources.

The Mission Generator is designed to be able to handle a variety of problems involving aircraft flights within the United States. Using the Mission Generator, one can construct a simulated network of flights between city pairs. This network could represent a fleet of vehicles operated by a specific air carrier or a collection of similar vehicles flown under different dispatch and routing conditions. An analyst can substitute one or more hypothetical vehicles for those in a base-case network to assess the impact of a new aircraft technology on flight resources.

The Mission Generator is designed to compute optimal trajectories for a given network of flights. This capability gives the analyst a means to investigate the impact of different route planning options (trajectory rules) for a given city pair or an entire set of air carrier operations (see Figure A-1). The Mission Generator includes the effects of winds in the computation of the optimal trajectories, and can detect special use airspace violations.

The Mission Generator calculates flight trajectory, time, and fuel consumption on a one flight segment at a time basis. It has the ability to automatically treat a variety of weather conditions (number of days of winds aloft data) for a number of flights.

*Figure A-1. Flight Planning Route Options*



## THE FLIGHT SEGMENT

At the core of the Mission Generator is its ability to optimize the in-air portion of a flight for minimum fuel under certain wind conditions. However, the Mission Generator is capable of producing data for a flight beyond just the optimized cruise phase of the flight.

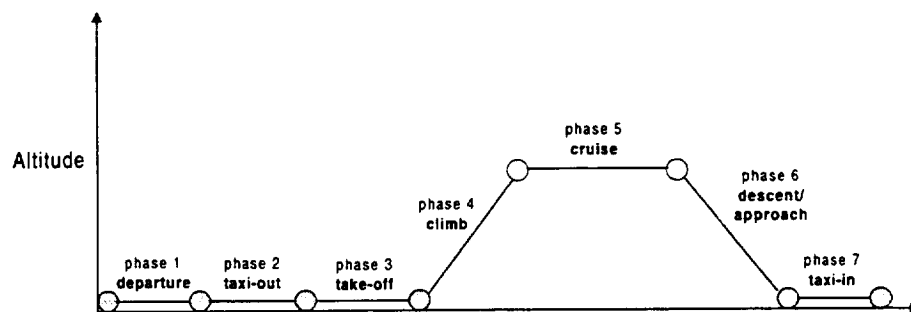
A real "flight," of course, consists of more than just time in the air. There is time at both airports, possibly time circling at the destination, etc. The Mission Generator describes each flight as composed of seven chronological phases:

- ◆ Departure                      at the gate waiting to taxi
- ◆ Taxi out                        to the takeoff runway

- ◆ Takeoff of the vehicle from the destination airport to 50 ft.
- ◆ Climb from 50 feet to cruise altitude
- ◆ Cruise steady flight to the destination
- ◆ Descent/approach descend, approach the destination airport, and land
- ◆ Taxi in from the primary runway to the gate.

Figure A-2 shows the seven phases.

*Figure A-2. The Phases of a Flight Segment*



Taken together, these phases constitute a “flight segment,” or just a “segment.” Furthermore, each phase has two parts: operation and delay. These parts are not physical parts of the phase; rather, they simply are divisions of the results data for the phase. The purpose of these parts is to delineate the results data between normal operations and delays (e.g., due to bad weather and busy times). Much of the delay data, especially, is looked up in files that hold historical data for various airports, airlines, and aircraft, etc.

To be able to accommodate increased knowledge of the aviation system as it relates to either models or data, each phase of flight is addressed in a separate module of the model and each in turn addresses the time and fuel needed to conduct the normal operation of a phase of flight and the time and fuel consumed due to delay in that phase of flight. Either data that drives phase of flight modules or the modules themselves can be easily upgraded as new data or other models become available. For example, one could provide the results of an airport delay model as a table of expected delays by airport and aircraft type to the module that deals with approach delays.

---

## RUNNING THE MODEL

The ASAC FSCM Mission Generator is coded in C++ and is hosted by ASAC on a UNIX server. Remote (and simplified) access is provided by the World Wide Web. Executing the model is simple.

Using Netscape™ or Internet Explorer™, go to the ASAC home page at <http://www/asac.lmi.org/>. From there, follow the pages until you get to the “ASAC model wizard.” From the model wizard, select ASAC Flight Segment Cost Model-Mission Generator, as shown in Figure A-3

*Figure A-3. Entering the ASAC FSCM Mission Generator via the World Wide Web*

**ASAC Model Wizard**

This will allow you to select a model or models to run, choose all appropriate options for the model or models chosen, then run the models and notify you of the results according to your selection.

Click [here](#) for help or complete documentation

Select a model to run  
then press **CONTINUE**:

ASAC Airport Capacity Model	↑ ↓
ASAC Airport Delay Model	
ASAC Flight Segment Cost Model - Mission Generator	
ASAC Flight Segment Cost Model - Cost Translator	
ASAC Air Carrier Network Cost Model	

**CONTINUE**

A user is required to provide only two sets of inputs, the list of flight segments to be flown and the particular rule to use to determine flight trajectories. All other settings, including pointers to default data files, are contained in a Scenario File that can be used as provided or modified in whole or in part by the model user (i.e., analyst).



## PREPARING MODEL INPUTS

The model user is required to provide the Mission Generator with one or more flight segments to be flown and the trajectory rule for flying those flights. Additionally, a variety of other data is available in a default library. The user may use this data, substitute his/her own files, or modify the default data for use in the analysis being conducted.

### The Network Description

The Mission Generator receives its primary input on flights to be flown and what period to fly them in a Network Description File (.NDF). You can use or modify an existing NDF or upload a new NDF via the Web interface shown in Figure A-4.

Figure A-4. NDF Entry

#### Mission Generator NDF File Entry Screen

The network description file (NDF) is a required file for running the Mission Generator Model. It describes each segment of a desired flight network, and the file as a whole represents the entire network. Each line of the NDF describes flights between a pair of cities, and these flights are a segment of a network.

Please enter the location of the NDF file below and press CONTINUE.

Network Description File:	<input type="radio"/> Upload New NDF	<a href="#">Help</a>
	<input checked="" type="radio"/> Find NDF File on Server	

To understand the NDF, consider a hypothetical new carrier—Far North Airlines—with flights between Alaska and regional cities. Far North also flies small aircraft to outpost towns and vacation getaway locations. Let's say that Far North's fleet is as shown in Table A-1.

*Table A-1. Far North Airlines Fleet*

Quantity	Aircraft
3	757
4	737
6	Falcon 50

Table A-2 consists of the flights Far North operates.

*Table A-2. Far North Airlines Flight Operations*

Origin	Destination	Flights per day	Seasons
Juneau	Seattle	2	all year
Fairbanks	Calgary	3	all year
Fairbanks	Seattle	2	all year
Fairbanks	Ketchikan	2	all year
Fairbanks	Nome	0.7	summer (no weekend flights)
Fairbanks	Anchorage	1	all year
Fairbanks	Juneau	1	all year
Fairbanks	Dutch Harbor	3	summer

The NDF for Far North Airlines (airline "FNA") is shown in Table A-3.

*Table A-3. Sample NDF for Far North Airlines*

ORIG	DEST	EQP	ARLN	START	END	DEP	ARR	BLK	FL/DAY
JNU	SEA	757	FNA	001	365	10:00	13:00	3:00	2
FAI	CAL	757	FNA	001	365	9:00	11:00	2:00	3
FAI	SEA	757	FNA	001	365	9:00	11:00	2:00	2
FAI	KTN	737	FNA	001	365	9:00	11:00	2:00	2
FAI	OME	FK50	FNA	135	240	9:00	11:00	2:00	0.7
FAI	ANC	737	FNA	001	365	9:00	11:00	2:00	1
FAI	JNU	757	FNA	001	365	9:00	11:00	2:00	1
FAI	DUT	Falcon	FNA	140	255	9:00	11:00	2:00	3

The first line in Table A-3 is a label, the fields of which stand for “origin airport, destination airport, equipment (aircraft), airline, starting date for the set of flights, ending date for the set of flights, scheduled departure time, scheduled arrival time, block time, and flights per day.” The data in the lines below the label line correspond with the fields in the label, and are in the same order.

Notice that the starting and ending dates are Julian dates, or day of the year. All flights that are flown all year have starting and ending dates of 1 and 365, respectively. The summer flights have starting and ending dates representing summer days only.

As the starting and ending dates imply, a given flight will be flown by the program for each day specified in that line of the NDF; however, the flight is flown only once per day, regardless of the number of flights per day specified. Repeating the flight on a given day would give the same results, but the same flight on differing days will give varying results due to varying wind conditions.

The departure and arrival times given are disregarded by the program (with one exception—Special Use Airspace), mainly because wind data is not broken down to time of day. Thus, the results of the flight will be the same regardless of the time of day. The block time, however, is important because it gives the scheduled elapsed time—the maximum time the flight *should* take. The Mission Generator uses the block time as an upper limit constraint while it is optimizing a flight.

The flights per day is an average. The 0.7 value for Fairbanks to Whitehorse is approximately 5/7, or 5 days flown out of a 7-day week – no weekend flights.

Note that each described flight—each line of the NDF—is “flown” by the program only in one direction—from the origin to the destination. To describe each round trip would take two lines in the NDF, with the origin and destination reversed, and the departure, arrival, and block times changed appropriately. Thus, the example NDF above describes only half the round trip flights.

Table A-3's sample NDF is in the correct format for the Mission Generator, and it could actually be input to the program if supporting data were available for the weather, airports, aircraft, and airline. For more detail about NDF and other input and output data formats, see Annex A-1.

---

## The Trajectory Rule

The trajectory rule is how the analyst tells the optimizer which of the four degrees of freedom, or design variables (if any) to find this optimal trajectory. The degrees of freedom are mach number (airspeed), altitude, latitude, and longitude.

There are six trajectory rules defined, as described below. Only the rules may be selected via the World Wide Web. Advanced users can work with the ASAC system administrator to add additional rules. Annex A-2 describes how advanced users may construct new trajectory rules.

1. **“great circle route (vary Mach number and altitude)”**

This rule optimizes the Mach number and altitudes for each waypoint on the route. The path points, latitude, and longitude are computed along the great circle and are not optimized. The figure of merit is defined as the fuel used in the cruise operation phase and it is minimized. The constraint is the cruise operation phase time, and it is specified to be less than the allowable phase block time. This allowable block time is computed by subtracting the total of all phases in the flight, except the cruise operation, from the specified block time in the Network Description File (NDF).

2. **“great circle route (vary Mach number and do step climb)”**

Step climb means that a flight may change altitude one time only during the cruise phase of flight. This rule optimizes the Mach numbers for each waypoint on the route and optimizes the two altitudes used when the step climb algorithm is active. The path points, latitude, and longitude are computed along the great circle and are not optimized.

The figure of merit is defined as the fuel used in the cruise operation phase and it is minimized. The constraint is the cruise operation phase time, and it is specified to be less than the allowable phase block time. This allowable block time is computed by subtracting the total of all phases in the flight, except the cruise operation, from the specified block time in the NDF.

3. **“free flight (vary path and Mach number and altitude)”**

This rule optimizes the Latitudes, Longitudes, Mach numbers, and altitudes for each waypoint on the route. The path points, latitude, and longitude are computed along the great circle as initial conditions.

The figure of merit is defined as the fuel used in the cruise operation phase and it is minimized. The constraint is the cruise operation phase time, and it is

specified to be less than the allowable phase block time. This allowable block time is computed by subtracting the total of all phases in the flight, except the cruise operation, from the specified block time in the NDF.

4. **“free flight (vary path and Mach number and do step climb)”**

This rule optimizes the Latitudes, Longitudes, and Mach numbers for each waypoint on the route and optimizes the two altitudes used when the step climb algorithm is active. The path points, latitude and longitude are computed along the great circle as initial conditions.

The figure of merit is defined as the fuel used in the cruise operation phase, and it is minimized. The constraint is the cruise operation phase time and it is specified to be less than the allowable phase block time. This allowable block time is computed by subtracting the total of all phases in the flight, except the cruise operation, from the specified block time in the NDF.

5. **“user-defined path (vary Mach number and altitude)”**

This rule optimizes the Mach number and altitudes for each user-defined waypoint on the route. The path points, latitude, and longitude are computed based on user-defined set of points found in the route database and are not optimized. UNIX users may download a utility to help them build this database (see Annex A-3) or they may simply follow the format described in Annex A-1 to build the database file.

The figure of merit is defined as the fuel used in the cruise operation phase, and it is minimized. The constraint is the cruise operation phase time, and it is specified to be less than the allowable phase block time. This allowable block time is computed by subtracting the total of all phases in the flight, except the cruise operation, from the specified block time in the NDF.

6. **“user-defined path (vary Mach number and do step climb)”**

This rule optimizes the Mach numbers for each waypoint on the route and optimizes the two altitudes used when the step climb algorithm is active. The path points, latitude, and longitude are computed based on user-defined set of points found in the route database and are not optimized.

The figure of merit is defined as the fuel used in the cruise operation phase and it is minimized. The constraint is the cruise operation phase time and it is specified to be less than the allowable phase block time. This allowable block

---

time is computed by subtracting the total of all phases in the flight, except the cruise operation, from the specified block time in the NDF.

Trajectory rules are summarized in Table A-4.

*Table A-4. Trajectory Rules Available*

Base rule	SUA checking	FAA altitude rule
Great circle route (vary Mach number and altitude)	X	
Great circle route (vary Mach number and do step climb)	X	X
Free flight (vary path and Mach number and altitude)	X	
Free flight (vary path and Mach number and do step climb)	X	X
User-defined path (vary Mach number and altitude)	X	
User-defined path (vary Mach number and do step climb)	X	X

In addition to these “base” trajectory rules, the user may add to the end of the trajectory rule two options that will affect the path decided upon by the optimizer. These two options are “SUA checking” and the “FAA altitude rule.”

The SUA (Special Use Airspace) checking option will take the optimized route and detect any violation restricted, prohibited, or warning airspace (by latitude and longitude). The detailed output of trajectories flown is modified to show the total number of SUAs violated, which spaces were violated, and whether or not those spaces are intermittent. Using this information, the user can perform a simple check on the validity of an optimized path. Note: The Mission Generator does not fly around SUAs since many of these are intermittent and/or frequently deactivated.

The FAA altitude rule option will force the optimized waypoint altitudes to conform to the FAA restriction of even thousands of feet on eastbound routes and odd thousands of feet on westbound routes as shown in Table A-5. This option is only available for trajectory rules that have specified a step climb.

Table A-5. FAA-Approved Altitudes

Eastbound (ft.)	Westbound (ft.)
19,000	18,000
21,000	20,000
23,000	22,000
25,000	24,000
27,000	26,000
29,000	28,000
33,000	31,000
37,000	35,000
41,000	39,000
45,000	43,000

## Other Inputs: The Scenario File

The Scenario File provides a simple way to track other model settings as well as substitute new, or modify existing, default values. It contains the following information:

- ◆ Optimizer settings (see model operation below)
- ◆ Location of waypoints database
- ◆ Location of master trajectory rules (not user-changeable)
- ◆ Weather (winds aloft) sampling rule and location of winds aloft data
- ◆ Known aircraft types and the location of aircraft performance data in the Base of Aircraft Data (BADA) format
- ◆ The location of aircraft and airline load factors information
- ◆ The location of phase of flight data for all phases of flight other than cruise
- ◆ Settings for use in any flight routing optimization.
- ◆ Location of output files (automatic via the World Wide Web).

Each of the file/directory locations in the Scenario File enable users to substitute completely new files for existing default values. If the users desire to modify existing or default data, they may do so via the World Wide Web interface, which then copies the modified data into a new file for use by the user. These files may then be saved for reuse at another time. Figures A-5 and A-6 illustrate the use of the World Wide Web interface for use with the Scenario File.

*Figure A-5. Scenario File Entry Screen*

**Mission Generator  
Scenario File  
Entry Screen**

The scenario file lists fundamental parameters needed for the Mission Generator program, and points to files of other data which give further detailed parameters. Only one scenario file is allowed per program run, and this file is required.

Please enter the location of the scenario file below and press CONTINUE.

Scenario File:	<input type="radio"/> Use Default Scenario File	Help
	<input type="radio"/> Upload New Scenario File	
	<input type="radio"/> Find Scenario File on Server	
	<input checked="" type="radio"/> Build New Scenario File	



Figure A-6. Building a Scenario File on the Scenario File Customizer Options Screen

**Mission Generator  
Scenario File  
Customizer Options**

You have chosen to customize the scenario file used with the Mission Generator using values from /opt/ASAC/Data/MG/default.sf. Please enter the name of your new scenario file, and select the categories of the scenario file you would like to modify.

Enter new scenario file name:  
[Text Input Field]

☐ Optimizer Values  
☐ Wind Data Files  
☐ Detailed Outputs  
☐ Lookup Files  
☐ Aircraft Manager Values  
☐ Weather Sampling Data  
☐ Lead Factors File

CONTINUE

## MODEL OPERATION

The Mission Generator operates conceptually on levels. At its most detailed level, it models one flight on one day. At the next level, the model operates on one flight over a period of time with a variety of wind conditions. Last, the model can repeat its operations for a variety of flights defined in an input network. Each of these is discussed below.

### Single Flight Modeling

This section provides additional detail about how the flight profile is modeled and how the optimization problem is prepared and executed. Where applicable, the simplifying assumptions made pertaining to the limitations of the model are explained. Also, additional information is provided about how to construct new trajectory rules and enhance the optimization problem solved by the Mission Generator.

---

## Vehicle Modeling

This section provides detail about how the aircraft model is created and how the individual disciplines (i.e., aerodynamic, propulsion, and weights) are represented.

### AERODYNAMIC AND PROPULSION MODELS

The lift and thrust characteristics of the flight vehicle are modeled using the Base of Aircraft Data (BADA) coefficients in a series of analytic equations. The BADA models were developed by Eurocontrol, the European Organization for the Safety of Air Navigation, at the organization's Eurocontrol Experimental Centre in Bretigny, France. Documentation of the models is provided by European Economic Community (EEC) Note 5/96, *User Manual for the Base of Aircraft Data (BADA), Revision 2.4*. This report is available from Eurocontrol Experimental Centre, Publications Office, B. P. 15, 91222—Bretigny-sur-Orge CEDEX, France. An Adobe Acrobat file (.PDF file) of the document is available for downloading at [www.eurocontrol.fr](http://www.eurocontrol.fr). The free-ware Adobe Acrobat Reader can be downloaded from [www.adobe.com](http://www.adobe.com).

### DRAG MODEL

The aircraft drag coefficient is computed from the BADA coefficients found in the aircraft data lookup file. These coefficients include:

$C_{D0}$  = zero lift drag coefficient

$C_{D2}$  = induced drag factor ("k")

$C_{D16}$  = transonic correction factor

The drag coefficient is given by

$$C_D = (C_{D0} + C_{D2}C_L^2)(10 + C_{D16}M^{16})$$

where  $C_L$  is the lift coefficient and  $M$  is the Mach number. The model was developed under the assumption that the lift is equal to the weight during each segment of the cruise phase. Also,  $C_{D0}$  and  $C_{D2}$  are assumed constant with altitude and Mach number.

The drag force at any point in the trajectory is computed in Newtons (N) from,

$$D = C_D \cdot qS$$

where  $S$  is the reference wing area ( $m^2$ ) and  $q$  is the dynamic pressure,  $N/m^2$ , given by

$$q = \frac{1}{2} \rho V^2$$

and  $\rho$  is the density ( $kg/m^3$ ) (a function of altitude) and  $V$  is the velocity (m/s), obtained from the segment Mach number ( $M$ ) and the local altitude speed of sound (a function of altitude).

## THRUST MODEL

The fuel-burn model used for the cruise phase of the trajectory is derived from the BADA formulation. This model utilizes coefficients obtained from the aircraft data lookup file. The equation used for calculation of the aircraft installed, total thrust in Newtons (N) is given for jet aircraft by

$$T = C_{t1} \left( 1.0 - \frac{\bar{h}}{C_{t2}} + \frac{C_{t3}}{\bar{h}^2} \right)$$

and for turboprop aircraft by

$$T = \frac{C_{t1}}{V} \left( 1.0 - \frac{\bar{h}}{C_{t2}} \right) + C_{t3}$$

where  $h$  is the average altitude computed between each segment endpoint, and  $C_{t1}$ ,  $C_{t2}$ , and  $C_{t3}$  are thrust factors from the BADA format. This thrust is the maximum thrust computed in N and assumes contribution from all engines.

The specific fuel flow (kg/hr./N) for the jet aircraft model is obtained from,

$$\eta = 0.06 \left( C_{f1} \left[ 10 + \frac{V}{C_{f2}} \right] \right)$$

and for the turboprop model by

---


$$\eta = 0.06 \left( \frac{V \cdot C_{f1}}{1000} \left[ 10 - \frac{V}{C_{f2}} \right] \right).$$

## WEIGHT COMPUTATION

The aircraft payload and fuel weights are handled by the following equations:

The computed payload weight is given by,

$$W_{Payload} = (n_{seats} W_{seat} l_{f_{pax}}) + (W_{cargo} l_{f_{cargo}})$$

where  $l_{f_{pax}}$  is the passenger load factor and  $l_{f_{cargo}}$  is the cargo load factor.

**Note:** We use 200 lbs./seat, which is for the passenger plus baggage.

The aircraft data lookup file has the maximum capacity for the payload weight and the fuel weight. If these weights are exceeded, a warning message appears in the output of the model.

A reserve fuel weight is also specified in the aircraft data lookup file. This reserve is added to empty weight before the mission begins and is used to occupy a reserve of the fuel capacity such that a diversion and extra 45 minutes of flight can take place on a typical 4-hour flight.

## TRAJECTORY MODELING

The method used to compute the fuel burn within the cruise phase is derived from a path segment analysis taking into account an energy correction for changes in altitude and speed.

First, given two points in space that represent the starting and ending points for a given segment on a flight path, we derive an expression for the fuel-burn for flight at constant altitude equal to the mean altitude of the path and constant airspeed equal the airspeed at the starting point of the path. Using the BADA fuel-flow rates, the rate of change of fuel is given by

$$\frac{dW}{dt} = -gF$$

where  $F$  is the rate of fuel flowing kg/hr.

Since velocity is assumed to be constant along this path, and that time (t) =d/V, we can write an expression for the change in weight per unit distance along the path,

$$\frac{dW}{dx} = \frac{-gF}{V_{KT}}$$

where  $V_{KT}$  is velocity in knots.

In unaccelerated, level flight

$$L = W$$

and

$$T = D.$$

Since,

$$D = L \left[ \frac{D}{L} \right].$$

it follows that,

$$D = W \left[ \frac{C_D}{C_L} \right].$$

In the BADA models,

$$\frac{C_D}{C_L} = \left( \frac{C_{D0}}{C_L} + C_{D2} C_L \right) (1 + C_{D16} M^{16})$$

since, in level flight,

$$C_L = \frac{W}{qS},$$

we have,

---


$$D = W \left( \frac{C_{D0} q_s}{W} + \frac{C_{D2} W}{q_s} \right) (1 + C_{D16} M^{16}).$$

Distributing weight and the Mach number drag-rise factor yields

$$D = C_{D0} q_s (1 + C_{D16} M^{16}) + \frac{C_{D2} (1 + C_{D16} M^{16})}{q_s} W^2.$$

The fuel flow (from BADA) is given by

$$F = \frac{dW}{dt} = T \eta$$

where  $h$  is the specific fuel consumption and  $T$  is the thrust. Substituting into  $dw/dx$  and replacing the thrust with the drag results in

$$\frac{dW}{dx} = \frac{\eta}{V} C_{D0} q_s (1 + C_{D16} M^{16}) + \left( \frac{\eta}{V} \frac{C_{D2} (1 + C_{D16} M^{16})}{q_s} \right) W^2.$$

This takes the form,

$$\frac{dW}{dx} = -a - bW^2.$$

The solution to this differential equation takes the form,

$$W_i = \sqrt{\frac{a}{b}} \tau \alpha v \left[ \tau \alpha v^{-1} \left( \sqrt{\frac{b}{a}} W_0 \right) - \sqrt{ab} (x_1 - x_0) \right]$$

where

$$a = \frac{\eta}{V} C_{D0} \cdot q_s (1 + C_{D16} \cdot M^{16})$$

and

$$b = \frac{\eta}{V} \frac{C_{D2} (1 + C_{D16} \cdot M^{16})}{q_s}.$$

We evaluated the weight of fuel burned on the segment as  $W_0 - W_1$ , corrected for any altitude and speed changes by the total energy method. We made that correction in this way: During flight at constant altitude  $h$  m and speed  $V$  m/s, the propulsion system delivers energy at the rate  $TV$  watts and fuel flows at the rate  $\approx T/3600$  kg/sec. If thrust were increased by  $\delta T$  for  $\delta t$  seconds, the propulsion system would deliver additional energy  $\delta E = \delta TV \delta t$  Joules, and consume an additional mass  $\delta f = \approx \delta T \delta t / 3600$  kg of fuel. Thus, the fuel  $\delta f$  required to produce energy change  $\delta E$  is given by

$$\delta f = \frac{\eta \delta E}{3600 V} \text{ kg}$$

Evaluating the energy change corresponding to altitude change  $\Delta h$  and speed change from  $V_0$  to  $V_1$  as

$$\delta E = \bar{W} \delta h + \frac{\bar{W}}{g} (V_1^2 - V_0^2)$$

where  $\bar{W} \equiv 0.5(W_0 + W_1)$ , we approximated the fuel burn on the segment as

$$W_f = W_0 - W_1 + g \frac{\eta \delta E}{3600 \bar{V}}$$

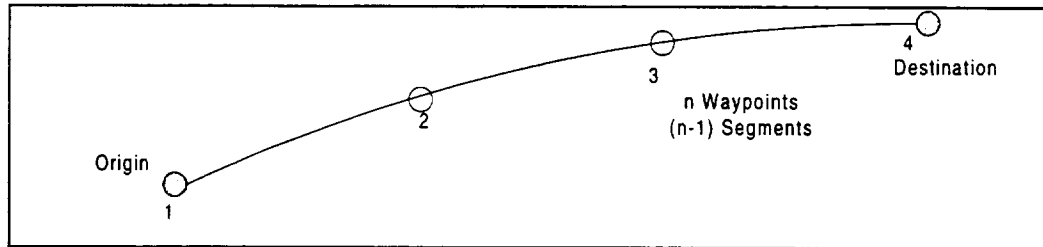
with  $\bar{V} \equiv 0.5(V_0 + V_1)$ .

## CRUISE-PHASE SEGMENT MODEL

The cruise model was developed such that a trajectory could be flown and only the necessary fuel weight for the given mission would be computed. This is accomplished by starting with only enough fuel for reserve flight time, then progressively computing fuel usage for each segment of the flight and adding that fuel weight to the total vehicle weight.

The flight segment is divided into  $N$  segments as shown in Figure A-7. For the Mission Generator model,  $N$  is set constant to 10 segments for each path.

*Figure A-7. Flight Path Segment between Two Cities Showing Individual Waypoints*



## PATH SEGMENT DETAILS

The wind is interpolated by using points at the midpoint between the defining waypoints. The Mach number used for the segment is that of the first waypoint.

The wind effects on the vehicle are computed by projecting the two-dimensional wind vector ( $w$ ) onto the segment path through

$$w_i = w\tau.$$

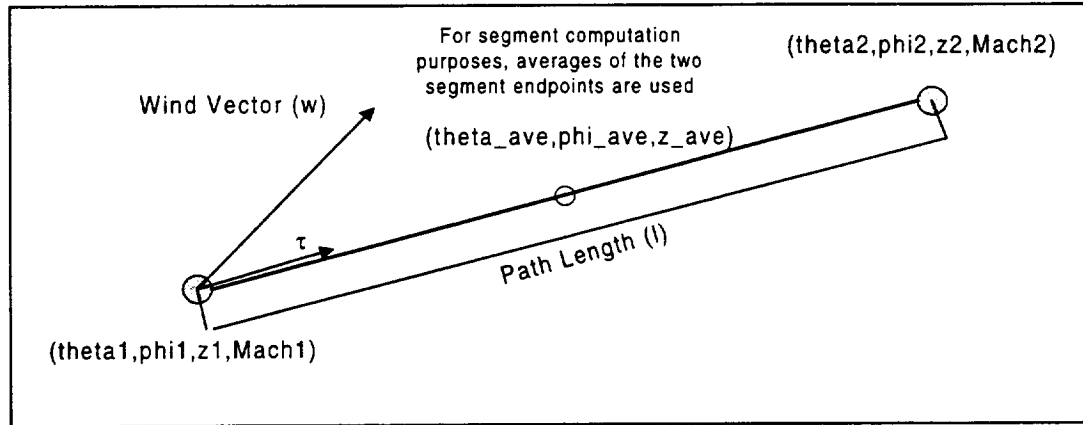
The speed along the segment is given by,

$$s = w_i + \sqrt{(V^2 - |w|^2 + w_i^2)}.$$



Figure A-8 illustrates the calculations.

Figure A-8. Segment Description Used Between Two Waypoints



Thus, using the segment speed, time can be computed for this subsegment by

$$t = l / s$$

and the distance that the airplane travels on the segment is given by

$$d = st.$$

## Initial and Final Waypoint Cruise Altitudes

The model has user-defined settings to begin the cruise portion of the flight profile, normally, 23,000 feet for jet aircraft and 10,000 feet for turboprop aircraft. This assumption was made after examining typical flight operations and when the terminal area controllers released the flight crew for en route control. The Mach number for this first cruise segment point is a design variable in the optimization problem. At the end of the cruise segment, the final altitude is a user-input value. The default for this ending altitude is 10,000 feet and the speed is limited to 250 kt. These values can be found in the individual aircraft input files, allowing the user to define different cruise start and end conditions for each aircraft, if desired.

---

## FUEL PENALTY IF THRUST IS LESS THAN DRAG

If the required thrust for the cruise segment is larger than the available thrust from the engine model, a penalty factor is incurred on the fuel coefficient. This penalty is an exponential function of the thrust deficiency, given by the difference of the drag and thrust:

$$cfuel = e^{.01*(D-T)}.$$

## WIND MODELING: INTERPOLATION OF THE MEASURED WIND DATA

The wind information is stored in separate binary files equivalent to the northward and eastward components of wind velocity at various pressure altitudes in millibars. The current set of files is given in Table A-6 below.

*Table A-6. Winds Aloft Data Points*

File name	Altitude (ft.)
200u	38,700
200v	38,700
250u	33,700
250v	33,700
300u	29,300
300v	29,300
400u	22,400
400v	22,400
500u	17,000
500v	17,000

Values for eastward and northward components of winds aloft were obtained from NOAA's National Center for Environmental Prediction (NCEP). The data were created in the Atmospheric Chemistry and Dynamics Branch of NASA/Goddard Space Flight Center.

The NCEP wind data are given on a grid, with 2° latitude increments and 5° longitude increments, from 22° to 52° north latitude and from 60° to 135° west longitude. NCEP wind data are available on constant pressure surfaces, and we used those for 500, 400, 300, 250, and 200 millibars. There is one data set for each day in calendar year 1995, except for January 20, July 11, and August 10; data were not available for those dates.

To model winds at arbitrary positions, we first interpolated with respect to altitude for both  $u$  and  $v$  values on the standard grid, using quadratic spline interpolation. We then used classical four-point interpolation to generate values of  $u$  and  $v$  at intermediate positions in the horizontal plane. (See Figure A-9 for an example of the winds at 38,000 feet on January 1, 1995.)

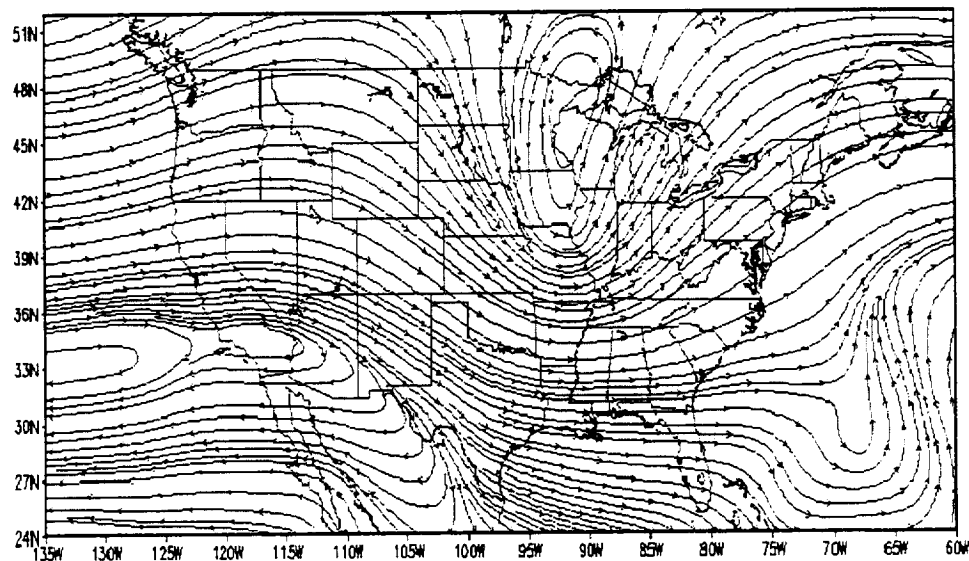
Winds are read from each binary file into a one-dimensional array and then are loaded into a three-dimensional array in the model. Each data point corresponds to a longitude, latitude, and altitude, and is stored in 240 four-byte element blocks per day.

The format for the binary data is a string of longitudes (16 elements) for each latitude (15 elements) in succession from the starting latitude to the ending latitude.

For example, the data are stored for each altitude file and direction component in the following typical string of data:

$V[\text{long}_1, \text{lat}_1] \ V[\text{long}_2, \text{lat}_1] \ \dots \ V[\text{long}_i, \text{lat}_1] \ \dots \ V[\text{long}_j, \text{lat}_j]$

*Figure A-9. Example of Wind Stream Lines at 38,000 Feet on January 1, 1995*



---

## GREAT CIRCLE ROUTE COMPUTATION

The model initializes the original path points using a great circle path. The model initializes the waypoint vector array for the optimizer by computing this path and setting each point to the segment points on the great circle. Each element of the route array is a vector containing three elements: The first two are the longitude and latitude, respectively, in radians, and the third is the altitude in nautical miles. The initialization keeps the first and the last route points at the values assigned by read-in data. Intermediate values are the coordinates of points on the great circle between the initial segment and the final. Altitude are set to a normal cruise value found in each aircraft description file.

## OPTIMIZATION PROBLEM

The optimization routine uses the information about the route, including Mach, altitude, latitude, and longitude to compute optimal flight profiles. The objective for the current set of trajectory rules is to minimize the fuel while meeting a specified time constraint. This constraint time is specified from the block time in the Network Description File.

## METHODS

DOT (design optimization tools) uses several algorithms for the numerical search routine to minimize the fuel. All are gradient-based, and utilize the functions of the vehicle modeling routines to obtain fuel and time values. The following options are available to the user through the "method" flag in the Scenario File.

- ◆ *Method 1* is the modified method of feasible directions (MMFD). It is a very efficient algorithm for computer memory; however, convergence to the time constraints might not be as accurate as in Method 3.
- ◆ *Method 2* is a sequential linear programming (SLP) method and is not recommended for the current suite of trajectory rules. The linear programming problem is useful when the design space, including the objective function and the constraint set, is linear. However, most of the vehicle and wind route modeling algorithms used in the mission generator are nonlinear and will not respond well to linear programming techniques.
- ◆ *Method 3* is the sequential quadratic programming (SQP) method. This is considered the best algorithm for convergence if the constraints are well placed; however, if the initial point begins in a highly infeasible region, the constraint might not be satisfied. This method is slightly more resource-intensive.

## INITIAL TRAJECTORY

The optimizer must have an initial set of data for the flight trajectory, which it then varies as it works toward finding the optimal route and speeds. Initial data must be given at each route waypoint for Mach number, altitude, waypoint latitude, and waypoint longitude.

The initial waypoint latitudes and longitudes are determined by defining the waypoints as evenly-spaced points along the great circle between the origin and destination airports. The capability to set the waypoints as those defined for a specific route between two airports is available as a utility program.

The initial Mach numbers are defined in each individual aircraft file (Mave). The initial altitudes are defined in each individual aircraft file (Have). These data were taken from average operating procedures from the Airline Procedures file for the BADA information.

## OPTIMIZATION PROBLEM

The parameter being optimized by the Mission Generator is aircraft fuel consumption, which must be minimized by the program. What varies in order to achieve this optimized value is the flight trajectory—that is, the path followed by the aircraft, plus its speed at different points along that path.

The trajectory is modeled by dividing the full path into line segments of roughly equal length. The points forming these line segments are called waypoints. At each waypoint there is expressed a latitude, longitude, altitude, and aircraft velocity. It is the interaction between the aircraft velocity and the wind velocity at these points that is analyzed by the optimizer, as it searches for the optimal route points and speeds between those points. Its general goal is to minimize headwind (or maximize tailwind) without excessively increasing the distance flown.

## SINGLE FLIGHT IN A VARIETY OF WINDS—WEATHER SAMPLING

### Single Flight Modeling

The user has the choice of two options in the Mission Generator for the execution of the described flight network: (1) every day specified in the NDF can be flown or (2) representative days can be flown via the definition of a frequency-weighted weather

---

histogram. Use of this sampling histogram choice should save run time, while giving results similar to the results obtained if every defined day had been flown.

The mode used is specified in the Scenario File under the “[Sampling]” tag. If all days in the date range specified in the NDF should be flown (every day mode), then the Scenario File should have<sup>1</sup>

```
[Sampling]
average_mode = FALSE
```

In this mode, every day in the Network Description File will be flown using representative winds aloft data from 1995. Statistical variance is given for both time and fuel on each line of results in the NRF. These variances are calculated using the grand total time and grand total fuel for each day flown; no variance is given for individual phase or delay/operation data.

If the program is run in “every day” mode, then all data points are available; thus, the variance is for the full population. In this case, the variance equation is

$$\text{variance} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

where  $n$  is the number of days run,  $\bar{x}$  is the mean of the data, and  $x_i$  is the value for a given day.

If only the days defined in the weather histogram should be flown (sampling mode), then the Scenario File should have

```
[Sampling]
average_mode = TRUE
sample_histogram = weather.hst
```

where weather.hst is the histogram file that will be used by the program.

The idea behind the use of the weather histogram is that days of the year with similar wind conditions could be represented by one of those days, which could then be flown and the results taken to be the results that would be obtained if all those similar days had been flown. Thus, the weather histogram is really a wind histogram – a way of condensing similar wind data into representative data. The wind data considered is

---

<sup>1</sup> We use this changed font to denote actual file contents and formats throughout this document.

wind speed and direction at various flight altitudes throughout the continental United States.

When using a weather histogram (in sampling mode), it is important to determine that the flights that will be flown are within the scope of the definition of that histogram. For example, say that the daily winds aloft data for the Northeastern region of the country showed that winds were largely unchanged for January 1 through January 6. Thus, one day could be defined in a weather histogram that could represent those 6 days. Say, however, that winds across other portions of the country were rapidly changing during this period. Thus, the weather histogram defined for the Northeast wind data would be applicable to flights within that region, but could lead to skewed results if it was applied to flights outside the region. The use of the weather histogram can be a useful tool for condensing data and saving program run time, but care should be taken in determining when it should be used.

Consider the following sample weather histogram file.

```
[Period_Information]; semicolons are comment
delimiter

Number_of_Periods = 2

Period = JAN
Period = FEB

[JAN]; Days are in Julian days
Period_Length = 001 031; Format = beginning date end
date
Day = 01.3; notice the day, then the weight
Day = 10.7

[FEB]
Period_Length = 032 059; February
Day = 035.4
Day = 045.6
```

A weather histogram is defined as a set of periods, and each period in turn has representative days defined, with a fractional weighting also defined for each day. The sum of the daily weights should be one for *each* defined period.

It is important to note that during sampling mode, the interaction between the weather histogram period definitions and the start and end dates in the line of the NDF can cause varying results. The reason is this: if at least one day in the specified range in a line of the NDF falls in a period defined in the weather histogram, then the *entire* pe-

riod in the histogram is flown. Thus, an NDF segment whose start date was close to the end of a histogram period, for example, would be run for that entire histogram period, which could deflect the results from those which would have been obtained by running in every day mode.

The weather histogram file format is designed to allow flexibility in how a flight network is represented. Periods do not have to be defined along month boundaries, for example. To avoid the histogram/NDF interaction problem described above, it is wise to define a weather histogram such that it “fits” the network described—the period of the year and region of the country to be flown.

If the program is run in “sampling” mode, then the variance calculation is more complicated, because it involves data within the defined periods in the weather histogram file, as well as possibly several periods crossed in order to run all the days specified in the NDF. For simplification, the variance equation when the program is in sampling mode is

$$\text{variance} = \frac{\sum_{i=1}^k f_i (x_i - \bar{x})^2}{1 - \frac{1}{n}}$$

where

$n$  is the total number of days in all the defined periods used for flight of the line of the NDF

$k$  is the total number of sampling days defined and used (i.e., the number of data points), and

$f_i$  is the total relative frequency of each day’s data, which is defined as

$$f_i = \frac{h_i}{p} w_i$$

where  $h_i$  is the frequency of that day within the period,

$p$  is the number of days in that period, and

$w_i$  is the relative (fractional) frequency of the period among all the periods used,

$x_i$  is the value for a given day, and

$\bar{x}$  is the frequency-weighted mean for all the data.



## Multiple Flight Modeling

Modeling multiple flight segments is accomplished by treating each flight (each input data row in the Network Description File) as independent. All functions for a single flight on one day, and on multiple days, is identical to what is described above.

## MODEL OUTPUT

Normal model output is in the form of a Network Resource File. Additional details are available in two additional file formats: a “raw output” with highlights of each individual day’s optimization for every flight and a “trajectory output” with detailed trajectory information for each day’s optimization for every flight.

### The Network Resource File

The results from the program execution of a flight network is produced by gathering statistics from individual flight analyses. Network results parallel the network description, with one line of results produced for every line of the description file. However, each of these lines of results is a summary of data from possibly many individual flights, according to specifications in the NDF.

The network results are posted to a file called the Network Resource File (NRF). This file contains three sections: (1) a program execution summary, (2) a listing of the Scenario Files used for the trial, and (3) the actual results from the flight of the network. A full description of the NRF is given under “Network Resource File” in Annex A-1 of this document. Below is a discussion of Annex A-1 of the file, the actual results.

The two parameters for which results are listed are time and fuel consumption. The unit for time is fractional hours (as opposed to hours and minutes; for example, a result of 2.5 hours would equal 2 hours, 30 minutes). The unit for fuel consumption is pounds. These two values are given for each phase of the flight *and* are further broken down into the values for the operation and delay parts of each phase. Total time and fuel consumption is also given for total delay of all phases, total operation of all phases, and a grand total for the entire flight.

It is important to note that each time and fuel result given is an average of the values for all the days flown for that flight segment description—in other words, for that line of the NDF. If weighting mode was not selected in the Scenario File, then these averages are simple averages of all the day results between and including the starting and ending dates specified in the line of the NDF. If weighting mode *was* selected, then

---

these averages are weighted averages using the weights for each day flown, as defined in the weather histogram.

Finally, the statistical variance is given for both time and fuel on each line of results in the NRF. These variances are calculated using the grand total time and grand total fuel for each day flown; no variance is given for individual phase or delay/operation data.

## Other Optional Output

After the optimizer has optimized the trajectory, the calculated fuel consumption and elapsed time—together with the waypoint latitudes, longitudes, and Mach numbers (speeds)—are written to two files of data. A line is written to these files for each day flown; no averaging or summarization is done. These files are not final results, but rather intermediate results which, if necessary, could be analyzed to explain some of the trends produced by the optimizer.

The names of these files are specified in the `raw_output` and `traj_output` fields under the “Path” tag in the Scenario File (usually, file names are `raw.out` and `traj.out`). A list of elements written by each file is shown below.

- ◆ `raw.out`:
  - NRF line name
  - day
  - great circle cruise phase time
  - great circle cruise phase fuel
  - block time available for the cruise phase optimization
  - optimized cruise phase time
  - optimized cruise phase fuel
  - takeoff weight
  - optimization error flag

- ◆ traj.out:
  - NRF line name
  - day
  - waypoint altitudes, listed serially from origin to destination
  - waypoint Mach numbers
  - waypoint longitudes, with 0 at the Greenwich meridian, and West longitudes as negative values
  - waypoint latitudes
  - number of Special Use Area's (SUA) violated (0 if SUA violation detection was not specified)
  - string specifying the specific SUA's violated (blank if SUA violation detection was not specified)

In addition to the optimizer processing, the cruise-phase fuel and time are calculated for a great circle flight. These fuel and time results are listed in the "raw\_output" file (usually raw.out) for comparison to the optimized trajectory results.

For the great circle route, the route is flown in the wind, as is the optimized route. Unlike the optimized route, the great circle route is flown under typical conditions (constant airspeed and fuel flows), and without a time constraint.

## SAMPLE RESULTS

Figures A-10 through A-14 show some sample results that have been obtained from runs of the Mission Generator program.

*Figure A-10. Average per-Flight Cruise Phase Fuel Usage for September for 13 Flight Segments*

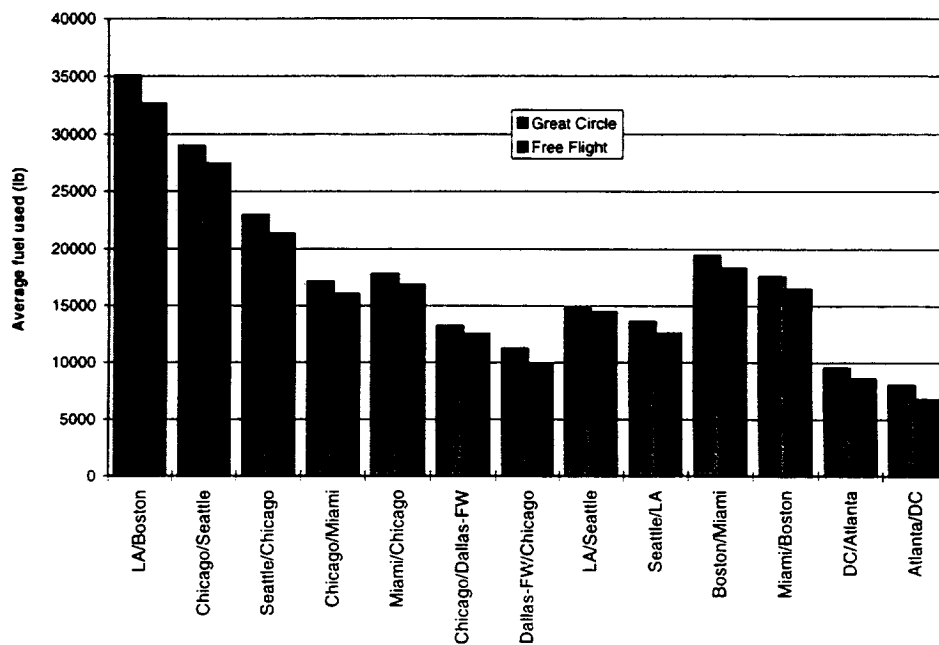


Figure A-10 shows the average fuel consumed per flight for runs on all days of September for 13 sample flight segments. Notice the fuel savings of the consistent free flight optimization trajectory versus the great circle route at fixed altitude and Mach number.

Figure A-11 shows the altitudes at all the sample flight segment waypoints for the first 20 days of the year. The figure demonstrates how the altitude varies in response to the wind field on those various days.

*Figure A-11. Altitude Profiles for Days 1 to 20 for a Sample Flight Segment with Mach/Altitude Optimization*

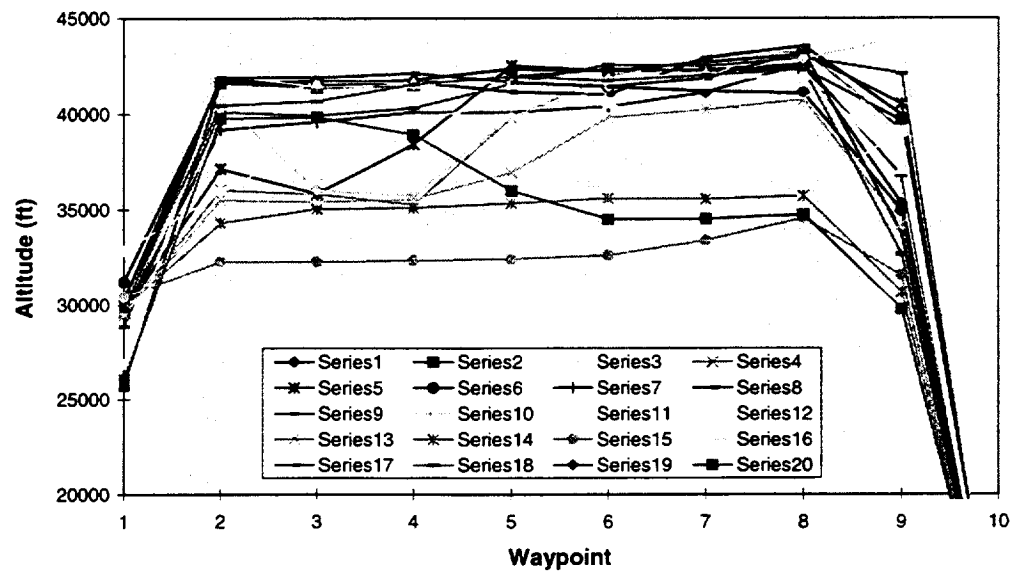


Figure A-12 shows how the waypoints move as a result of optimization under two different trajectory rules. Notice that for the great circle trajectory rule, the waypoint latitudes and longitudes are fixed at the values determined for them along the great circle route. For the free flight trajectory rule, however, the latitudes and longitudes clearly move off the great circle initial route. The data reflect a flight between Los Angeles and Boston on the 294<sup>th</sup> day of the year.

*Figure A-12. Comparison of Waypoints for Two Trajectory Rules*

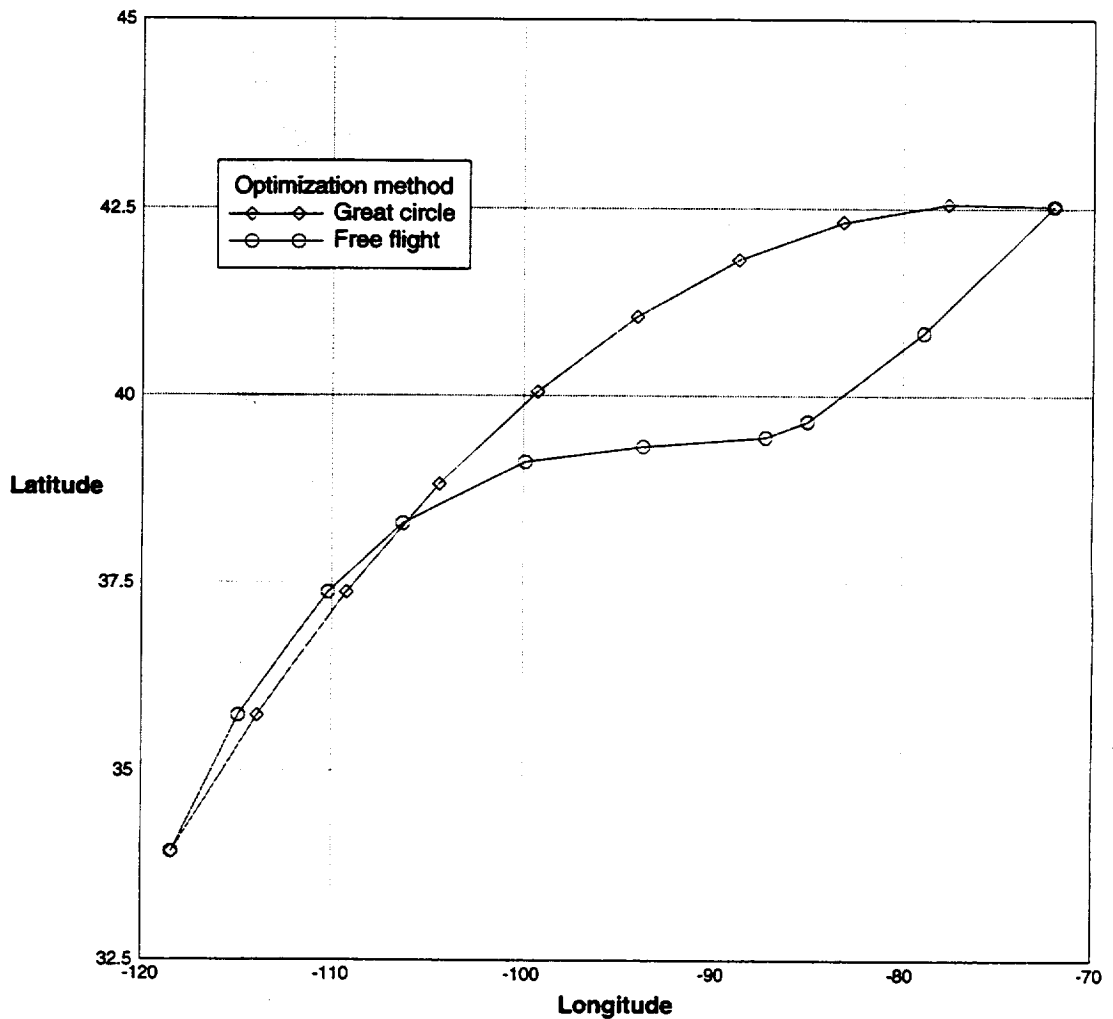


Figure A-13 shows the great circle and optimized routes for a flight between Los Angeles and Boston. Wind stream lines also are shown. Notice how the optimized path tries to stay in the regions of high wind (the general direction of the wind is west to east) to take advantage of the tailwind.

*Figure A-13. Comparison of Great Circle and Optimized Routes,  
Shown with a Wind Field*

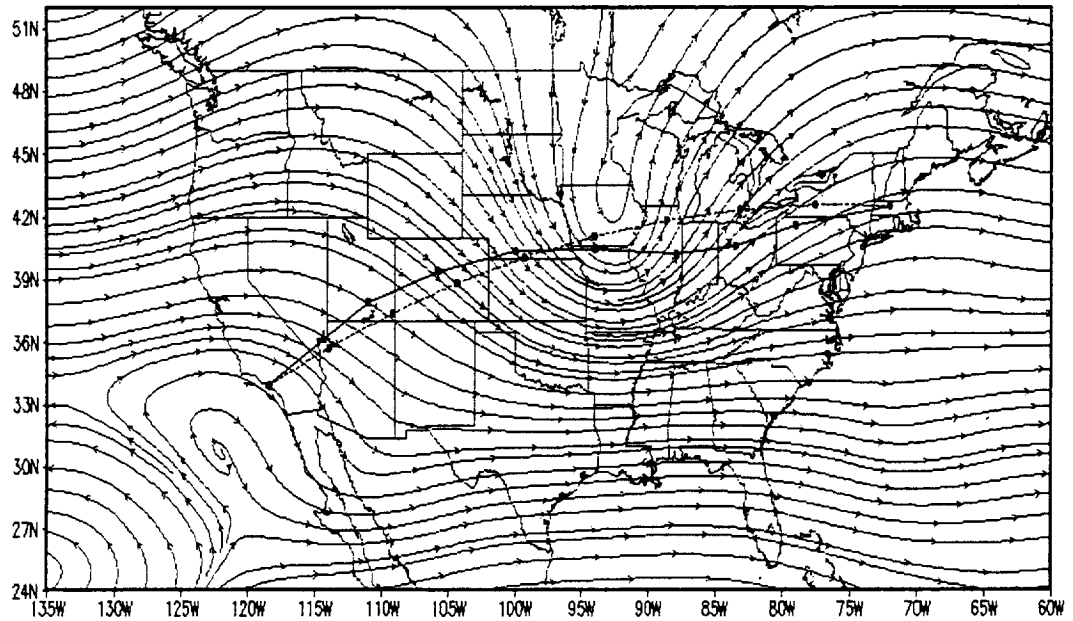
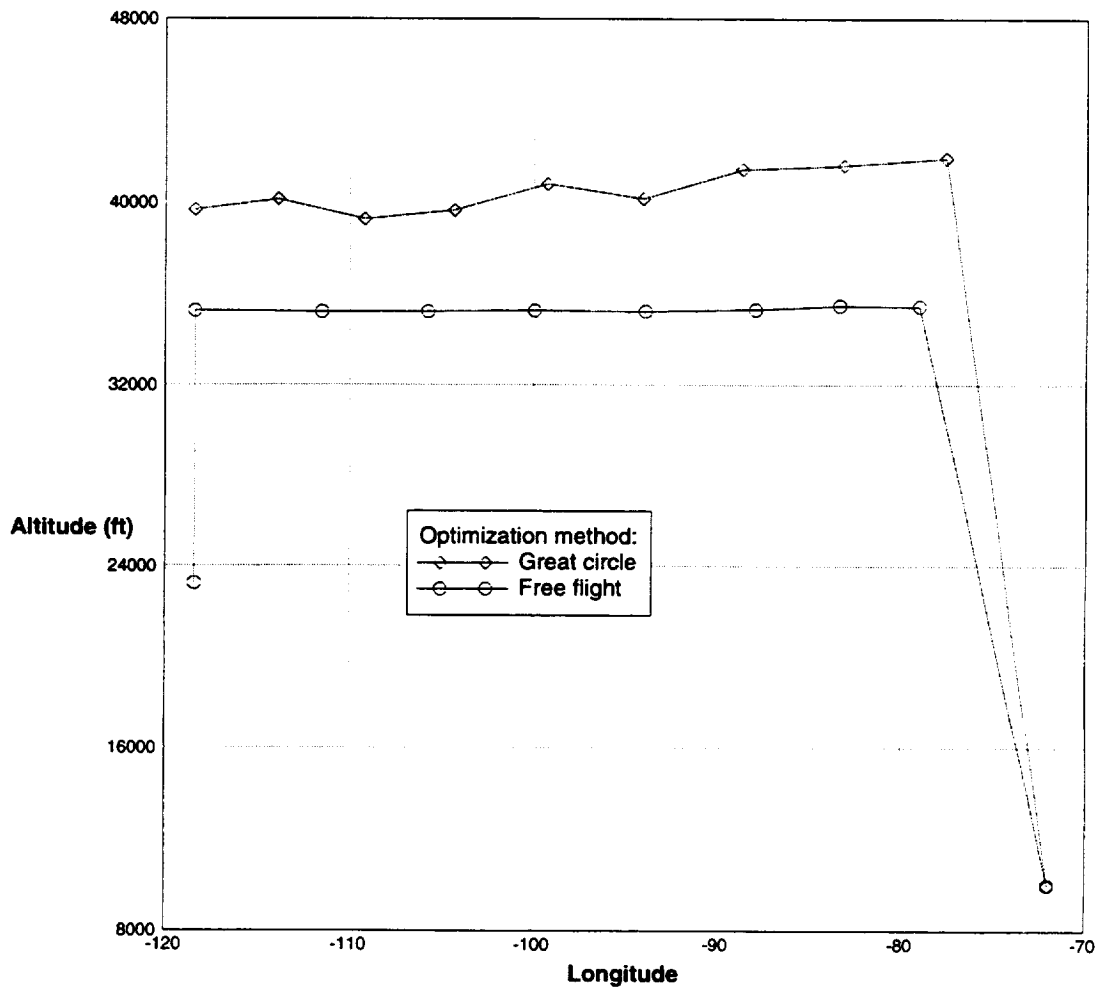


Figure A-14 shows the difference in altitudes resulting from optimization via two defined trajectory rules for the same flight. Notice again how the waypoint longitudes move only for the trajectory rule that enables such movement.

*Figure A-14. Comparison of Altitudes for Two Trajectory Rules*





# Annex A-1

## ASAC Flight Segment Cost Model Mission Generator

### File Formats

The Mission Generator file formats are described in the sections below.

## NETWORK DESCRIPTION FILE

This file is an input file uploaded by the user via the World Wide Web.

### Purpose

To describe the flight segments—between pairs of cities—of a flight network.

### Description

The network description file describes each segment of a desired flight network. The file as a whole represents the network. The record format is:

```
<origin> <destination> <aircraft> <airline> <start  
date> <end date> <departure time> <arrival time> <block  
time> <flights/day factor>
```

Each line of the file (“line of NDF”) describes flight(s) between a pair of cities; these flights are a segment of a network. All the segments together – i.e., all of the lines of the file – constitute the network. Each city pair may occur more than once in the file because (1) multiple airlines and aircraft fly the route, (2) multiple flights (during a day) may occur, and (3) the line is dependent upon direction; so, two lines are needed for a round-trip.

A description of each record (line) is contained in Table A-1-1.

*Table A-1-1. Record Line Descriptions*

Origin	Three-character flight destination airport symbol
Destination	Three-character flight destination airport symbol
Aircraft	Aircraft name; this name should match exactly an aircraft name in the Aircraft Name List File; if the name used includes a space, then the entire name must be enclosed in double quotes (quotes are optional otherwise)
Airline	Two-character airline symbol
Start date	Julian day of year on which to start flying the segment
End date	Julian day of the year on which to stop flying the segment
Departure time (GMT)	In the form HH:MM (hours:minutes); 24-hour clock
Arrival time	In the form HH:MM (GMT)
Block time	In the form HH:MM; the time allowed for the complete flight
Flights/day factor	A real number representing the number of flights per day; a fractional value indicates that the flight is not flown every day (e.g., not flown on weekends)

An example line of NDF follows:

```
LAX   BOS   757   UA    015   030   07:30   13:00
5:30   0.8
```

How the segment is actually “flown” by the program depends upon the mode specified in the Scenario File: every day mode or sampling mode. Every day mode will actually “fly” the flight on each day specified in the line of the NDF file. Sampling mode will use the weather histogram to fly selected days (based on weather), and then weight those days to represent the entire histogram period.

## Leap Year

The wind data employed is for 365 days—one full year. However, in a leap year, flights will be flown on the 366<sup>th</sup> day, also. Since no wind data are available for this day, the wind from day 365 will be used for flights on day 366. Statistics will be collected for day 366, just as is done for any other day of the year.

The definition of day 366 is allowed in the NDF file *only* in leap years. The year 2000 is a valid leap year.

## SCENARIO FILE

### Purpose

To provide or point to data that is needed to run the Mission Generator. This is a “master” file that provides—directly or indirectly—most of the input data needed for the Mission Generator program.

### Description

The Scenario File lists fundamental parameters needed for the Mission Generator program, and points to files of other data that give further detailed parameters. Only one Scenario File is allowed per program run, and this file is required.

The file consists entirely of a series of tags, under which fields and associated values give the needed parameters. The general tag format is

```
[<tag>]
<field> = <value>
<field> = <value>
etc.
```

```
[<tag>]
etc.
```

The possible tags and fields are (with example values)

```
[Opt_Controller]
convergence_iterations = 3;
delobj= 1.2e-4
scale_flag =TRUE
dot_scale_flag= 1
method= 3
dot_output_level = 0

[Path]
DB_PATH = ./pref_routes
PROJECT_PATH = ./projects
home = /home/LMI/FSCM/files
etc.
```

---

```
[Cruise_Operations]
WIND_DATA = WIND_DATA
raw_output = /home/raw/raw.out
traj_output = /home/raw/traj.out

[Departure_Operations]
file_name = LOOKUP/departure_operation.dat

[Departure_Delay]
file_name = LOOKUP/departure_delay.dat

[Note: The two departure lookup files, above, are repeated for all flight phases, except cruise operations. Please see "Trajectory Phase Lookup Files" below for details.]

[Aircraft_Manager]
ac_index = "index.ac"
NEW1 = "Boeing 7x7"
NEW1_file = "7x7.dat"

[Sampling]
average_mode = FALSE
sample_histogram= weather.hst;

[Load_File]
load_file_name = LOOKUP/load_factors.dat
```

## Explanation of File Tags<sup>1</sup>

### [Opt\_Controller]

**convergence\_iterations** indicates the maximum number of times the optimizer should iterate to try to get a better solution. Normally, a low number (two to four) is all that is required.

**delobj** is the delta in the objective function which is "close enough". This threshold value tells the optimizer to stop iterating when changes are less than the specified amount. The default for this value is  $1.2e^{-4}$ .

---

<sup>1</sup> A tag is a section label that identifies the data that follows.

**scale\_flag** indicates whether or not the design variable parameters should be scaled before being passed to the optimizer. Enter “TRUE” or “FALSE.” The Typical setting for this flag is “TRUE.”

**dot\_scale\_flag** indicates whether or not the DOT optimizer should scale values with each optimization iteration. Set to 1 if scaling should be done; set to -1 if scaling should not be done. Typical setting for this flag is -1.

**method** gives the optimization method that should be used. Method alternatives are as follows:

- ◆ 1 means Modified Method of Feasible Directions (MMFD); this is the default.
- ◆ 2 means Sequential Linear Programming (SLP).
- ◆ 3 means Sequential Quadratic Programming (SQP).

**dot\_output\_level** gives a means for changing the raw output level from the optimizer (Design Optimization Tools or “dot”). A level of 0 gives no raw optimizer information; a level of 7 gives the maximum information. Typically, this raw optimizer information will not be needed by the user, but is available if necessary. The typical value is 0.

**“Path”** The fields defined under the Path tag are paths that can then be used in front of file names in other fields in the Scenario File. For example, a user could define

```
MINE = /user/george/lookups
```

and then use this path definition for some files

```
[Departure Operations]
file_name = $MINE/departure_operation.dat

[Departure Delay]
file_name = $MINE/departure_delay.dat
```

The rules for defining and using the path fields are as follows:

#### *Defining*

1. A valid UNIX path must be entered; an absolute or any relative path is allowed.

- 
2. If a path is not found, then the path will be changed to the program's current path.

#### *Using*

1. A dollar sign (“\$”) must be appended before the path name and placed before the desired file name (similar to UNIX environment variable usage).
2. A path name is allowed only at the beginning of the string specifying the file; it is not allowed in the middle of the name; for example, the following is not valid:

```
[Departure Delay]
file_name = /home/$MINE/departure_delay.dat
```

3. The path name in the file string must terminate with a slash, a blank, or an end-of-line character.

There are two special path names that must be used for their respective files, if a path is desired for those files. These paths are defined as usual. However, they are not used in front of a file name elsewhere in the Scenario File, as other paths are used. Rather, the paths are used directly for the files being used, as listed below. The two paths and the files that they point to are

DB_PATH	the route database; the file searched for in this path is Pref_DB.pag
PROJECT_PATH	the project file, which contains optimizer definitions for the trajectory rules; the file searched for in this path is fscm.proj

#### **[Cruise\_Operations]**

**WIND\_DATA** gives the path to the *directory* holding the wind data files. A path name defined under the [Path] tag may be placed at the beginning of this path definition.

**raw\_output** is the output file containing calculated time and fuel for each day flown, plus some other calculated flight values. The full path, including the file name, should be specified.

**traj\_output** is the output file containing latitude, longitude, altitude, and Mach number data at trajectory waypoints, as well as SUA violation data, for each day flown. The full path, including the file name, should be specified.

## NETWORK RESOURCE FILE

This file is the results output file. Its name is given by the -NRF option on the command line, or by default is the NDF (input) file name with the final extension replaced with “.NRF” (for example, default.nrf).

### Purpose

To list the results of the flight of a network (as described in an associated NDF file).

### Description

The network resource file (NRF) lists results for the flight of a network as it was described in a network description file (NDF). Each line (record) in the NRF parallels a line in the NDF input file. For example, if the input NDF had 30 lines (flight segments) defined, then the NRF also will have 30 lines of results.

Every NRF has three sections, which are separated by a line of equal signs (“=”). The three sections are (1) program execution summary, (2) Scenario File used, and (3) results of the program run (the flying of the network). These three sections contain the following:

- ◆ **Section 1 – program execution summary** is the date of the program run; the machine ID and other machine information for the machine on which the program ran.
- ◆ **Section 2 – Scenario File used** is the Scenario File which was used as input to the program is listed in its entirety.
- ◆ **Section 3 – results of the program run** is a line written describing the format of the results records to follow; then one long line of results is written for each line in the NDF for which flights were conducted (normally, all lines in the NDF are flown, unless there are preventing errors).

Each results record contains many values. The values are separated by single tab characters and the line is terminated, as usual, with a line feed. The format of each line of results is shown below; remember that this format is shown as the first line of Section 3.

```

SEG_NAME ORIG DEST EQUIP AIRLN START END DEP ARR BLK
FL/DAY   LOAD_FACTOR GROUND_STATUTE_MILES PH1_OP_TM
PH1_OP_FL PH1_DL_TM PH1_DL_FL PH1_TOT_TM PH1_TOT_FL
PH2_OP_TM PH2_OP_FL PH2_DL_TM PH2_DL_FL PH2_TOT_TM
PH2_TOT_FL PH3_OP_TM PH3_OP_FL PH3_DL_TM PH3_DL_FL
PH3_TOT_TM PH3_TOT_FL PH4_OP_TM PH4_OP_FL PH4_DL_TM
PH4_DL_FL PH4_TOT_TM PH4_TOT_FL PH5_OP_TM PH5_OP_FL
PH5_DL_TM PH5_DL_FL PH5_TOT_TM PH5_TOT_FL PH6_OP_TM
PH6_OP_FL PH6_DL_TM PH6_DL_FL PH6_TOT_TM PH6_TOT_FL
PH7_OP_TM PH7_OP_FL PH7_DL_TM PH7_DL_FL PH7_TOT_TM
PH7_TOT_FL TOT_OP_TM TOT_OP_FL TOT_DL_TM TOT_DL_FL
TOT_MNTM TOT_VRTM TOT_MNFL TOT_VRFL

```

*Note:* Spaces in the above description represent single tabs separating the values.

Table A-1-2 shows the abbreviations used.

*Table A-1-2. Abbreviations for Line Results*

Abbreviation	Description
SEG_NAME	segment name; this is a concatenation of the origin airport, the destination airport, the start date, and the end date; it serves as a quick reference to the results record
ORIG	origin airport
DEST	destination airport
EQUIP	aircraft
AIRLN	airline
START	start date
END	ending date
DEP	departure time
ARR	arrival time
BLK	block time
FL/DAY	flights per day



Following the above-listed terms, the average times and fuel consumptions for the flight are listed. The descriptions of these terms use the following additional abbreviations, shown in Table A-1-3, which are concatenated together, separated by underscores (“\_”), to form the descriptions of the terms.

Table A-1-3. Additional Abbreviations for Line Results

PH	Phase
OP	operations
DL	delay
TM	time
FL	fuel
TOT	total
MNTM	mean time
VRTM	variance time
MNFL	mean fuel
VRFL	variance fuel

## AUXILIARY FILES

### Aircraft Name List (Index) File

#### SCENARIO FILE TAG AND FIELD

```
[Aircraft_Manager]
ac_index = <your file>    (typical file is index.ac)
```

#### PURPOSE

To list aircraft names and associated data files. Also allowed under this tag is a list of aircraft names and data files, which are added at run-time to the list in the aircraft names file.

#### DESCRIPTION

The aircraft name list file is used to match aircraft names to the aircraft data file-names. The aircraft name list file should be edited whenever a new aircraft data file is to be used. (As an alternative, new aircraft files can be entered directly into the Sce-

---

nario File; see “Additional Method” below.) Each line contains an aircraft name and the data file for that aircraft, separated by a colon (:). Example:

```
Aircraft Name      : datafile.dat
"Boeing 757"       : b757.dat
"Boeing 767"       : b767.dat
"Airbus A320"      : ea320.dat
"Airbus A330"      : ea330.dat
```

The aircraft name will be listed in the third field of each line of the NDF file and the data file for that aircraft is found via this aircraft name list file.

As shown in the example above, there can be any amount of space (spaces or tabs) on either side of the colon, but there should not be any space before the aircraft name. There may be blank lines in between aircraft. The combination of the aircraft name and the whitespace before the colon should be no more than 80 characters long. The data filename also should be no more than 80 characters long, and it must be a valid filename. If an aircraft name has a space in it, then the entire aircraft name should be enclosed in quotes.

Comments are allowed on each line of the file, following a semicolon (;). All data following the semicolon will be disregarded, until a new line is encountered.

#### ADDITIONAL METHOD TO ADD AIRCRAFT FILES

Only one aircraft name list file can be given in the `ac_index` field in the Scenario File. However, additional aircraft files can be added individually, directly in the Scenario File, if so desired. The format is

```
<aircraft name 1 field> = "<aircraft name 1>"
<aircraft name 1 field>_file = <aircraft data file 1>
<aircraft name 2 field> = "<aircraft name 2>"
<aircraft name 2 field>_file = <aircraft data file 2>
etc.
```

These files should be listed in the Scenario File under the “[Aircraft\_Manager]” tag, immediately after the “`ac_index =`” line. An example:

```
ac_index = index.ac
NEW1 = "Boeing 7x7"
NEW1_file = 7x7.dat
NEW2 = "experimental"
NEW2_file = exp.dat
```

The “\_file” ending to the aircraft name field is required exactly, in order for the file to be retrieved. This method works in addition to the standard method of naming one index file.

## Aircraft Data File

This file is accessed via a reference in the Aircraft Index File.

### PURPOSE

To list aircraft-specific data needed for program calculations.

### DESCRIPTION

Each aircraft data file contains data on a single aircraft. For example, the file B757.ac could contain data for a Boeing 757. These data are taken from the Base of Aircraft Data (BADA) files, and used in the BADA models to calculate thrust, drag, fuel flows, and other needed parameters. New aircraft data files can be added to analyze new aircraft; in this case, the aircraft name and data file name should be added to the Aircraft Index File or the Scenario File (see “Aircraft Name List (Index) File” above).

The aircraft data file consists of 13 lines of data. The spacing of the values on each line is not important, as long as at least one space separates the values. The format of this file is

```
Vmo Mmo hmo
Vstall Vave Mave have
hstart hend Mend
OWE Wpayload Wfuel Swing ac_type
CD0 CD2 CM16
Cf1 Cf2 Cf3 Cf4
CT1 CT2 CT3
Cidle
Cmax comm
Cmax continuous
Ccruise lo
Ccruise hi
Capproach
```

The first line consists of the maximum operating velocity, the maximum operating Mach number, and the maximum operating altitude.

The second line consists of the cruise stall velocity, the cruise velocity at average mass level, the cruise Mach number at average mass level, and the average cruise altitude, which is taken to be 80 percent of the maximum operating altitude.

The third line consists of the altitude at which cruise begins (23,000 feet for jets, 10,000 feet for turboprops), the altitude at which cruise ends (set at 10,000 feet for all aircraft), and the ending Mach number (set at 0.392 for all aircraft).

The fourth line consists of the operating empty weight (OWE), the maximum payload weight, and the reserve fuel weight (lbs.); next is the wing area ( $m^2$ ). The final value on this line is the aircraft type, where 1=Jet and 2=Turboprop. The total takeoff weight of the aircraft is the sum of OWE, the reserve fuel, a portion of the maximum payload weight (as limited by varying load factors), and the weight of the calculated required fuel.

In the following definitions,

T = thrust  
h = height  
M = Mach number  
v = velocity  
r = air density  
S = wing area

The fifth line of the file is the BADA drag parameters, which are related as follows:

$$\begin{aligned} \text{total drag coefficient} &= C_D = (C_{D0} + C_{D2}C_L^2) (1 + C_{M16}M^{16}) \\ \text{drag} &= D = C_D r v_{TAS}^2 S / 2 \end{aligned}$$

The sixth line is the BADA fuel flow coefficients, which are related as follows:

$$\begin{aligned} \text{idle fuel flow rate for jet/turboprop} &= C_{f3} (1 - h/C_{f4}) \\ \text{idle fuel flow rate for piston} &= C_{f3} \\ \text{operational fuel flow for jet} &= C_{f1} (1 + v_{TAS} / C_{f2}) T \\ \text{operational fuel flow for turboprop} &= C_{f1} (1 - v_{TAS} / C_{f2}) (v_{TAS} / 1000) T \\ \text{operational fuel flow for piston} &= C_{f1} \end{aligned}$$

The seventh line is the BADA thrust parameters, which are related as follows:

$$\begin{aligned} \text{jet: } T_{\max \text{ climb (ISA)}} &= C_{T1} (1 - h/C_{T2} + C_{T3}/h^2) \\ \text{turboprop: } T_{\max \text{ climb (ISA)}} &= C_{T1} (1 - h/C_{T2}) / v_{TAS} + C_{T3} \\ \text{piston: } T_{\max \text{ climb (ISA)}} &= C_{T1} (1 - h/C_{T2}) + C_{T3} / v_{TAS} \end{aligned}$$

The remaining lines are various fuel flow parameters, in units of lbs./hour:

The eighth line is the aircraft idle fuel rate.  
The ninth line is the maximum commercial power fuel rate  
The tenth line is the maximum continuous power fuel rate  
The eleventh line is the cruise fuel rate 1 (at low mass)  
The twelfth line is the cruise fuel rate 2 (at high mass)  
The thirteenth line is the approach flight phase fuel rate

Comments are allowed at any point in the file, following a semi-colon (";"). Comments are terminated by the end-of-line character.

Annex A-4 explains of how to use either ACSYNT or FLOPS to generate the data for this file.

## Weather Histogram File

### SCENARIO FILE TAG AND FIELDS

```
[Sampling]
average_mode = TRUE
sample_histogram = <your file>
(typical file is weather.hst; default is default.hst)
```

### PURPOSE

To describe a frequency-weighted weather histogram.

### DESCRIPTION

The weather histogram file describes a series of periods and days within those periods. These definitions are used when the Mission Generator is run in sampling mode, which uses representative days to simulate a flight through many days. The purpose of the sampling mode is to save program run time.

---

The periods and days are defined in a specific format of tags and fields, as follows:

[Period\_Information]  
Number\_Of\_Periods =

<number of defined periods>      \*the number of periods  
defined in tag below;  
this number must be cor-  
rect for all of the peri-  
ods to be read correctly

Period <number>=<period name 1>    \*"number" must be a whole  
number; "period name" can  
be any string, but must  
be contained in double  
quotes if the string is  
more than one word

Period <number>=<period name 2>

Period <number>=<period name 3> etc.

[<period name>]      \*this      "period      name"  
matches one of those de-  
fined above

Period\_Length <start date>  
<end date>

\*the dates are Julian  
dates; e.g., Jan. 1 = 1,  
Dec. 31 = 365 (not a leap  
year), Feb. 1 = 32

Day = <date> < weight>

\*"number" is a Julian  
date; "frequency" is a  
real number representing  
the weight of that day in  
the period, and should be  
less than or equal to 1

Day = <date> <weight> etc.

[<period name 2>]

Period\_Length <start date> <end date>

Day = <date> <weight>

Day = <date> <weight>  
etc.

[<period name 3>]  
etc.

The [Period\_Information] tag and “Number\_Of\_Periods” field must be the first and second elements in the file, respectively.

Any number of days can be defined within a period, as long as it is at least one and not more than the number of days defined for the period. A day can be any integer 1 through 365, although 366 is allowed during a leap year (see below).

The sum of the weights for the days in each period must total 1; a tolerance of 0.001 is allowed on this total. Be careful with repeating number fractions (e.g.,  $1/3 = 0.333$ ) on the total. For example:

Day = 1.33  
Day = 10.33  
Day = 20.33

would fail the total of 1.0 check. One of the days should be 0.34, in this case.

Comments are allowed at the end of each line, following a semicolon (;). All data after the semicolon is disregarded until a new line is found.

## LEAP YEAR

The wind data employed is for 365 days – one full year. However, in a leap year, flights will be flown on the 366<sup>th</sup> day, also. Since no wind data are available for this day, the wind from day 365 will be used for flights on day 366. Statistics will be collected for day 366, just as is done for any other day of the year.

The definition of day 366 is allowed in the weather histogram file *only* in leap years. The year 2000 is a valid leap year.

---

## EXAMPLE

Below is an example of a Weather Histogram File:

```
[Period_Information]    ; semi colons are comment
delimiter

Number_Of_Periods = 3

Period 1 = "JAN" ;The quotes are only necessary is you
have more than one word

Period 2 = FEB

Period 3 = " The rest of the Year"  These names are the
names of the periods

["JAN"]                ; Days are in Julian days
Period_Length 001 031  ; Format = Beginning date end
date
Day = 010 1.0          ; Format = day relative
frequency

[FEB]
Period_Length 032 059  ; February
Day = 035 .3333
Day = 045 .7777

[" The rest of the Year"] ; This must appear exactly
as it is                  ; defined in the
period name
Period_Length 060 365
Day = 100 .5
Day = 200 .5
```



## Trajectory Phase Lookup Files

### SCENARIO FILE TAGS AND FIELDS

These are the Scenario File tags and fields:

```
[Depart_Operations]
file_name = departure_operation.dat

[Depart_Delay]
file_name = departure_delay.dat

[Taxi_Out_Operations]
file_name = taxi_out_operation.dat

[Taxi_Out_Delay]
file_name = taxi_out_delay.dat

[Takeoff_Operations]
file_name = takeoff_operation.dat

[Takeoff_Delay]
file_name = takeoff_delay.dat

[Climb_Operations]
file_name = climb_operation.dat

[Climb_Delay]
file_name = climb_delay.dat

[Cruise_Delay]
file_name = cruise_delay.dat

[Approach_Operations]
file_name = approach_operation.dat

[Approach_Delay]
file_name = approach_delay.dat

[Taxi_In_Operations]
file_name = taxi_in_operation.dat

[Taxi_In_Delay]
file_name = taxi_in_delay.dat
```

---

File names given above are typical, but you may substitute your own file names.

## PURPOSE

To provide time values for the “fixed” segments of the trajectory. These files are accessed similarly among them by the software, although the key varies (e.g., origin airport, and aircraft) depending upon the specific file.

## DESCRIPTION

There are seven phases in the trajectory, with each phase having operations and delay parts. Thus, there are 14 total parts, and the time values for 13 of these parts are found in these lookup tables. The fuel values are deduced from the time values. For the cruise operations part, the time and fuel values are calculated using the cruise optimizer. The time values in all the files are given in hours.

The tags for the lookup files—listed above—describe the 13 parts of the trajectory that uses these files. The lookup keys for each file are listed in Table A-1-4.

*Table A-1-4. Lookup Keys for Trajectory Files*

departure operations	origin airport
departure delay	origin airport
taxi-out operations	origin airport and airline
taxi-out delay	origin airport and airline
takeoff operations	aircraft
takeoff delay	origin airport
climb operations	aircraft
climb delay	origin airport
cruise operations	NOT A LOOKUP
cruise delay	destination airport
approach operations	destination airport
approach delay	destination airport
taxi-in operations	destination airport and airline
taxi-in delay	destination airport and airline

An example lookup file (portion) for taxi out delay as follows:

ABE	UA	0.119213
ABQ	UA	0.164168
ACK	UA	0.218421
ALB	UA	0.152454
AMA	UA	0.122575
ANC	UA	0.191057
ANI	UA	0.107011
ATL	UA	0.302034

Data for the lookup files was taken from the 1995 Airline Service Quality Performance (ASQP) data. These data are accessible at the World Wide Web site: [www.asac.lmi.org](http://www.asac.lmi.org).

## Aircraft Load Factors File

### SCENARIO FILE TAG AND FIELD

The Scenario File tag and field are

```
[Load_File]
load_file_name = <your file>
(typical file is load_factors.dat)
```

### PURPOSE

To provide aircraft load factors for each possible airline/aircraft combination.

### DESCRIPTION

This file provides three load factor parameters for each airline/aircraft combination specified. Only one load factor file is accepted. The record format is

```
<airline> <aircraft> <mail/cargo load factor> <seats>
<passenger load factor>
```

“Airline” should be a two-character airline symbol; “aircraft” should be a valid aircraft with an associated aircraft file as defined in the aircraft index file; “mail/cargo load factor” and “passenger load factor” should each be a real number less than one and greater than 0; “seats” should be the integer number of seats on the aircraft.

---

The two load factors are each a percentage of maximum weight, expressed as a fraction. When the program is calculating the aircraft takeoff weight, the load factors are multiplied against their respective portions of the maximum payload weight of the aircraft, with the appropriate values used for the airline and aircraft specified in the NDF. The “seats” value is used to determine the portion of total payload weight used for passengers, with the remainder being used for mail and cargo.

The following formulas describe the load factor/weight relations:

$$\text{total\_payload\_capacity} = \text{passenger\_payload\_capacity} + \text{MC\_capacity}$$
$$\text{passenger\_payload\_capacity} = (\text{number\_of\_seats}) (\text{weight\_per\_seat})$$
$$\text{actual\_payload} = ([\text{passenger\_payload\_capacity}] [\text{passenger\_LF}]) + ([\text{MC\_capacity}] [\text{MC\_LF}])$$

**Notes:** MC means “mail/cargo”; LF means “load factor”; the total payload capacity is a known constant value for a given aircraft; the weight per seat is 200 pounds, which is for the passenger plus baggage.

A sample file:

TW	757	0.65	130	0.672415
UA	757	0.65	130	0.663835
UE	757	0.65	130	0.516
US	757	0.65	130	0.660
UT	757	0.65	130	0.673
VP	757	0.65	130	0.672412
VS	757	0.65	130	0.673
WN	757	0.65	130	0.598
WR	757	0.65	130	0.673
YX	757	0.65	130	0.665519

## User-Specified Routing Input Data File

The input file needed for either creation of a new database or appending to an existing database should be in the format described below. The entire description represents one line of data in the file.

<route number><origin airport symbol> <destination airport symbol> <number of waypoints><point 1 name> <point 1 latitude> <point 1 longitude> <point 2 name> <point 2 latitude> <point 2 longitude> etc.

The waypoint data grouping is

<point n name> <point n latitude> <point n longitude>

This pattern repeats for each waypoint, up to and including the final waypoint. The total number of waypoint data groupings given should match the “number of waypoints” field listed earlier in the line of data.

**Note:** There is no space between the route number/origin airport symbol and number of waypoints/point name pairs of data. For these pairs, the two fields are delimited as the numerical and nonnumerical parts. All other fields are separated with at least one space; more than one space can separate these fields, however.

Following is an example of a file containing three routes from Atlanta—to Baltimore/Washington International (BWI), to Washington National (DCA), and to Denver (DEN). Each route is specified in the file as one line (three lines total are shown below).

1ATL	BWI	10ATL	33.6200	84.4300	SPA	35.0300
81.9200	GSO		36.0460	79.9760	10623	36.1210
79.8440	13238		37.0230	78.2180	CREWE	37.0970
78.0810	13148		37.2300	77.8330	13541	37.4940
77.3350	RIC		37.5020	77.3210	BWI	39.1700
76.6700						
1ATL	DCA	10ATL	33.6200	84.4300	SPA	35.0300
81.9200	GSO		36.0460	79.9760	10623	36.1210
79.8440	13238		37.0230	78.2180	CREWE	37.0970
78.0810	13148		37.2300	77.8330	13541	37.4940
77.3350	RIC		37.5020	77.3210	DCA	38.8500
77.0300						
1ATL	DEN	10ATL	33.6200	84.4300	VUZ	33.6700
86.8800	10582		33.9760	87.5550	MEM	35.0620
89.9810	RZC		36.2300	94.1200	ICT	37.7450
97.5830	GCK		37.9190	100.7240	HGO	38.8150
103.6260	IOC		39.4360	104.3370	DEN	39.8000
104.8870						



## Annex A-2

### Constructing a New Trajectory Rule

---

The trajectory rules are simply identifiers for different optimization problems that the user can ask the Mission Generator to perform. The trajectory rule specifies the design variables, objective function, and constraint set that the optimization routine will access. New trajectory rules can be added by editing the fscm.proj file. This file is the project controller for the optimizer. It is not modifiable over the World Wide Web. Advanced users may contact the ASAC system administrator to add new trajectory rules to this file.

The format and general procedure for adding a new trajectory rule is given in the next section.

#### GENERAL FORMAT FOR A TRAJECTORY RULE

The following format is an example of the input format for the project file (fscm.proj) and how to create additional trajectory rules.

```
Rule Name
n :Number of Design Variable Parameters
Parameter 1
Lower Limit  Upper Limit  Initial Value
Parameter 2
Lower Limit  Upper Limit  Initial Value
.
.
Parameter n
Lower Limit  Upper Limit  Initial Value
n :Number of Constraint Parameters
Constraint 1
Lower Limit  Upper Limit  Initial Value
Constraint 2
Lower Limit  Upper Limit  Initial Value
.
.
Constraint n
Lower Limit  Upper Limit  Initial Value
Objective Function Name
0 :Number of DV in history
```

---

```
0 :Number of DV Iterations.
0 :Number of Constraints

0 :Number of Con Iterations.
0 :Number of Merit Iterations.
```

```
Long/Min Fuel
8 :Number of Parameters
x1
-120 -50 -74
x2
-120 -50 -74
x3
-120 -50 -74
x4
-120 -50 -74
x5
-120 -50 -74
x6
-120 -50 -74
x7
-120 -50 -74
x8
-120 -50 -74
1 :Number of Parameters
Mission Time
-1.e15 4.5 5.0
Mission Fuel
0 :Number of DV in history
0 :Number of DV Iterations.
0 :Number of Constraints
0 :Number of Con Iterations.

0 :Number of Merit Iterations.
```

- ◆ The first line is the rule name. This is the trajectory rule name
- ◆ The next line is the number of parameters, i.e., design variables. This should be equal to the number of parameter specifications included in the next sections. For example, if you have eight design variables, enter 8 on this line.
- ◆ The next *n* lines follow the format for the parameter information. Each parameter is a set of two lines, the first for the name and the second for the numerical information. The name can be selected from a list of available parameters for the current version of the program. The next line after the *n*



parameter sections is the number of constraint parameters.

For example, if there is one constraint (time, etc.) enter 1 here. After the line specifying the number of constraint parameters, a section of parameters is specified for the constraint functions. This format is exactly the same as the design variables.

- ◆ After all the constraints are specified, the next line is the specification for the objective function (or merit function). This can be selected from the list provided in the next section.
- ◆ Finally, a series of history information must be entered. Since there is never a history for a new rule, simply copy the section of zeros as found in the example input file.

## What Is a Design Variable in the Project File?

A design variable is a parameter in the project file that the optimizer uses to control the objective function and constraint values. In the project file, the area shown by

```
Parameter 1
Lower Limit  Upper Limit  Initial Value
```

This means that Parameter 1 is the name of the variable to use as the design variable.

For example, the following parameter section specifies that Altitude 2 be used as a design variable, and that the limits for that altitude be set to 20,000 feet and 50,000 feet (See Figure A-2-1).

```
alt1
20000  50000  35000
```

*Figure A-2-1. Physical Description of Design Variables*

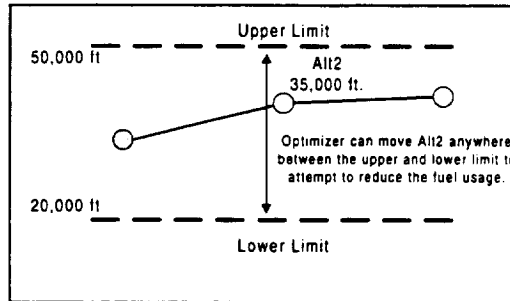


Table A-2-1 shows available design variable parameters for use in trajectory rule construction. These are the pieces of the cruise-phase trajectory that the optimizer will vary.

*Table A-2-1. Available Design Variable Parameters for Trajectory Rule Creation*

Phase	Longitude	Latitude	Altitude	Mach
1	x1	y1	alt1	mach1
2	x2	y2	alt2	mach2
3	x3	y3	alt3	mach3
4	x4	y4	alt4	mach4
5	x5	y5	alt5	mach5
6	x6	y6	alt6	mach6
7	x7	y7	alt7	mach7
8	x8	y8	alt8	mach8
9	x9	y9	alt9	mach9
10	x10	y10	alt10	mach10

Table A-2-2 shows available objective function variables also available as constraint functions for trajectory rule construction. Design values also may be limited in constraints.

*Table A-2-2. Available Objective Functions Variables for Trajectory Rules*

Name	Description
Mission fuel	Fuel used in the cruise portion of the flight
Mission time	Block time remaining available for cruise



## Annex A-3

# Route Database Utility

---

The route database utility is used to create user-specified route points between city pairs. When the “user-defined” trajectory rules are employed, the Mission Generator uses the points defined in the route database for the path. No optimization is performed on these points in the user-defined trajectory rules; therefore, these points remain fixed throughout the entire analysis. Creation and update of the route point database is possible by using the *prefDBUtil* utility available for download to Sun, Hewlett Packard, and Silicon Graphics UNIX platforms.

This program can be used to create a new route point database, or to append new data to an existing database. The program is primarily intended to hold route data and can be used to enter any set of waypoints that a user may wish to use for an experimental flight. Each record in the database holds the latitude and longitude of each waypoint along a particular route. Each route is limited to 55 waypoints total, and if the user wishes to use this route within the optimization problem, the route must have 10 waypoints with two of those points always being the origin airport and the destination airport.

The key for each record is the origin/destination combination. For example, a flight from Los Angeles (LAX) to Boston (BOS) would use the record in the database for LAXBOS. The reverse-direction route may have differing points, and thus would use the record found for BOSLAX. In other words, the route found is dependent on the direction of the flight.

The *prefDBUtil* utility uses UNIX’s ndbm database management system to take a file of route points (waypoints) and load them into a database file. Use of the database system should make access to a particular route much faster as compared to a flat file lookup. Note that the ndbm system does *not* guarantee locking on data that is being updated. Thus, when updating a database, it is wise to have any users of the Mission Generator logged off. However, since the Mission Generator accesses the database strictly for reading (not writing), the fast and widely-available ndbm system is sufficient for, and applicable to, the route data.

---

## PROGRAM OPERATION

When the *prefDBUtil* utility runs, it immediately asks

Would you like to create a new database? (yes/no)

“no” is the default answer and it is used if an invalid answer is given or <return> is pressed with no answer.

A “yes” answer indicates that the user wants to create new database files. The program then asks for the name of the input file of route data. Enter the full path and file name of this file. **WARNING:** If you type “yes” you will overwrite the existing route database.

The new database files will then be created. These files will always be called

```
Pref_DB.dir  
Pref_DB.pag
```

The files will be created on the directory on which the program is run. Any route containing more than 55 waypoints will *not* be added to the database; and a warning of such will be given.

If a “no” answer is given to the “create new database?” question, then the program prompts the user for a file of route data to append to the database. If a valid file is given, then the data will be appended to the database. If a duplicate route is found in the database, then a warning displays, and the existing data are overwritten with the data from the input file for that route.

If an invalid append file name is specified, then an error message displays, and the program stops.

## USER-SPECIFIED ROUTING INPUT DATA FILE

The input file needed for either creation of a new database or appending to an existing database should be in the format described below. The entire description below represents one line of data in the file.

```
<route number><origin airport symbol> <destination air-  
port symbol> <number of waypoints><point 1 name> <point  
1 latitude> <point 1 longitude> <point 2 name> <point 2  
latitude> <point 2 longitude> etc.
```

## The waypoint data grouping

<point n name> <point n latitude> <point n longitude>

pattern repeats for each waypoint, up to and including the final waypoint. The total number of waypoint data groupings given should match the “number of waypoints” field listed earlier in the line of data.

*Note:* There is no space between the route number/origin airport symbol and number of waypoints/point name pairs of data. For these pairs, the two fields are delimited as the numerical and nonnumerical parts. All other fields are separated with at least one space; more than one space can separate these fields, however.

Following is an example of a file containing three routes from Atlanta—to Baltimore/Washington International (BWI), to Washington National (DCA), and to Denver (DEN). Each route is specified in the file as one line (three lines total are shown below).

1ATL	BWI	10ATL	33.6200	84.4300	SPA	35.0300
81.9200	GSO		36.0460	79.9760	10623	36.1210
79.8440	13238		37.0230	78.2180	CREWE	37.0970
78.0810	13148		37.2300	77.8330	13541	37.4940
77.3350	RIC		37.5020	77.3210	BWI	39.1700
76.6700						
1ATL	DCA	10ATL	33.6200	84.4300	SPA	35.0300
81.9200	GSO		36.0460	79.9760	10623	36.1210
79.8440	13238		37.0230	78.2180	CREWE	37.0970
78.0810	13148		37.2300	77.8330	13541	37.4940
77.3350	RIC		37.5020	77.3210	DCA	38.8500
77.0300						
1ATL	DEN	10ATL	33.6200	84.4300	VUZ	33.6700
86.8800	10582		33.9760	87.5550	MEM	35.0620
89.9810	RZC		36.2300	94.1200	ICT	37.7450
97.5830	GCK		37.9190	100.7240	HGO	38.8150
103.6260	IOC		39.4360	104.3370	DEN	39.8000
104.8870						





## Annex A-4

# Converting ACSYNT Output to Mission Generator Aircraft Format

A sample aircraft data file for the Mission Generator looks like this:

```
;B737.ac
;Vmo      Mmo      Hmo
350      0.84      37000
;Vstall    Vave      Mave      Have
142      280      0.72      29600
;Hstart    Hend      Mend
23000      10000      0.392
;owe      wpayload    wfuel      swing      jet/turbo
66000      35120.8      5882.1      91      1
;cd0      cd2      cd16
0.0175      0.047726      2
;cf1      cf2      cf3      cf4
0.69629      425.72      14.7      63000
;ct1      ct2      ct3
95500      51000      4.15e-11
;Fuel rates
1940.4
11853.6
9482.88
4290
5662.8
1016.4
```

This section outlines how to find the necessary information to create a custom aircraft data file using the output information from either the Aircraft Synthesis or Flight Optimization System synthesis program.

---

## CREATING AN AIRCRAFT DATA FILE USING ACSYNT INFORMATION

Parameters listed in the aircraft file—an input file to the Mission Generator—are taken from BADA (Base of Aircraft Data) coefficients. (See “Aircraft Data File” documentation.) These BADA coefficients, in turn, are related to data in an ACSYNT (Aircraft Synthesis) file. Described below is the relationship between ACSYNT data and BADA coefficients, which could be used to create a mission generator aircraft file from an ACSYNT data file. Only the relationships needed to describe the aircraft file parameters are given. See the aircraft file documentation for a list of the parameters and brief descriptions.

### PERFORMANCE AND VEHICLE LIMITS

Name: **Vmo**

Description: Maximum operating velocity (kt)

Location: Derived

Details: Using the Mmo value found in the *envelop* input namelist derive Vmo from Mmo and the selected maximum altitude for the velocity.

Example: The following is an example of

```
$DATENV
```

```
COMFUL = .6, IDEBUG = 0, IVN      = 1, MLIMIT  
= 0.84, ...
```

Name: **Mmo**

Description: Maximum operating Mach number

Location: Input file

Details: Found in the *envelop* input namelist.

Example: The following is an example of

```
$DATENV
```

```
COMFUL = .6, IDEBUG = 0, IVN      = 1, MLIMIT  
= 0.84, ...
```

Name: **Hmo**

Description: Maximum operating altitude (ft)

Location: Output file

Details: The value for the maximum altitude for 1G, steady, level flight can be computed using the envelope routine in ACSYNT. Table A-4-1 is an example of the maximum altitude of 37,000 ft. computed for a max operating altitude as given by ACSYNT.

```
1.00G Flight Envelope, Specific Excess Power = .0:
```

*Table A-4-1. Recognizing Maximum Altitude*

Low/High Altitude	Mach Number	CL Comments	Mach Number	Q/M comments	Limit	Limit
0.	.145	CL LIMITED	.687	Q LIMITED	.145	.687
3300.	.154	CL LIMITED	.730	Q LIMITED	.154	.730
6600.	.204	CL LIMITED	.777	Q LIMITED	.204	.777
10000.	.218	CL LIMITED	.829	Q LIMITED	.218	.829
13300.	.233	CL LIMITED	.840	MACH LIMITED	.233	.840
16600.	.249	CL LIMITED	.840	MACH LIMITED	.249	.840
20000.	.267	CL LIMITED	.840	MACH LIMITED	.267	.840
23300.	.845	THRUST LIMITED	.840	MACH LIMITED	.285	.840
26600.	.842	THRUST LIMITED	.840	MACH LIMITED	.306	.840
30000.	.515	THRUST LIMITED	.833	THRUST LIMITED	.333	.840
33300.	.587	THRUST LIMITED	.826	THRUST LIMITED	.383	.840
35000.	.727	THRUST LIMITED	.820	THRUST LIMITED	.411	.840
36000.	.753	THRUST LIMITED	.805	THRUST LIMITED	.420	.840
37000.	.778	THRUST LIMITED	.801	THRUST LIMITED	.430	.840
38000.	.000	ABOVE THE FLIGHT ENVELOPE			.440	.840
39000.	.000	ABOVE THE FLIGHT ENVELOPE			.450	.840
40000.	.000	ABOVE THE FLIGHT ENVELOPE			.458	.840
41000.	.000	ABOVE THE FLIGHT ENVELOPE			.471	.840
42000.	.000	ABOVE THE FLIGHT ENVELOPE			.482	.840
43000.	.000	ABOVE THE FLIGHT ENVELOPE			.499	.840

Name: **Vstall**

Description: Cruise stall velocity (kt)

Location: Derived from Mstall

Details: Using the thrust-limited low-Mach number found in the 1-G envelope for the averaged cruise altitude. This is the used by the optimizer as a lower limit on the Mach number design variables. As a result, this variable should be selected to give a suitable working range for all altitudes that the optimizer might choose.

Name: **Vave**

Description: Cruise velocity at average mass cruise point (kt)

Location: Output file

Details: Trajectory output provides data for the cruise velocity. CAUTION: this output is in ft/s. Be sure to convert this variable to kt.

Name: **Mave**

Description: Cruise Mach at average mass cruise point

Location: Output file

Details: Trajectory output provides data for the cruise Mach.

Name: **Have**

Description: Average cruise altitude. (ft.)

Location: Output file

Details: Trajectory output provides data for the average cruise altitude.

Name: **Hstart**

Description: Cruise starting altitude. (ft.)

Location: USER DEFINED / Output file

Details: This is the altitude the user wishes the airplane to start the entire cruise phase. The selected altitude should be chosen to reflect where the climb segment ends and the cruise segment begins. An alternative to user input is selecting the end of the climb phase in the ACSYNT trajectory output file. This variable should default to 23,000 ft. for jet, and 10,000 ft. for the turboprop.

---

Name:       **Hend**

Description:   Cruise ending altitude. (ft.)

Location:      USER DEFINED / Output file

Details:       This is the altitude the user wishes the airplane to end the entire cruise phase. The selected altitude should be chosen to reflect where the cruise ends and the descent segment begins. An alternative to user input is selecting the altitude at the beginning of the descent phase in the ACSYNT trajectory output file. Usually, this variable defaults to 10,000 ft.

Name:       **Mend**

Description:   Cruise ending Mach

Location:      USER DEFINED / Output file

Details:       This is the Mach number the user specifies for which the airplane ends the entire cruise phase. This Mach should be chosen to reflect entry speed into the descent and landing phase. Typically this is chosen to be 0.392 to reflect the 250 kt speed limit at and below 10,000 ft in the terminal operating area. An alternative to user input is selecting the Mach at the beginning of the descent phase in the ACSYNT trajectory output file.

Figure A-4-1. Trajectory Output

PHASE	M	H	CL	ALPHA	WFUEL	TIME	VEL
SFC(I)	THRUST(I)	CD	GAMMA	W	WA	Q	
SFC(U)	THRUST(U)	CDINST	L/D	THR/THA	PR	X	
WARM-UP		0.			366.1	5.80	
1.02	3703.						
TAKEOFF	.16	0.	3.4489	14.53	784.1	2.00	177.
	.30	77989.	.8517	3.71	252553.6	1357.97	37.
	.30	77989.	.0024	4.05	1.00	1.00	3186.
2ND SEG	.16	400.	3.4489	14.53			177.
	.30	38994	.8517	5.17	252553.6	1357.97	37.
	.30	77989	.0024	4.05	1.00	1.00	
	{Hstart}						
CLIMB	.59	23000.	.7062	6.89	3142.0	11.28	608.
	.43	28365.	.0515	1.59	248627.6	671.22	212.
Cycle	.42	28585.	.0005	13.70	.87	1.00	59.
	{Mave}	{Have}					{Vave (ft/s)}
CRUISE	.80	36000.	.5444	4.58	23683.4	196.00	775.
	.45	15172	.0365	.00	224944.1	453.11	213.
Cycle	.45	15324.	.0004	14.93	.72	1.00	1500.
	{Mend}	{Hend}					
DESCENT	.59	23000.	.4855	4.56	132.6	6.84	608.
	-1.00	-139.	.0323	3.82	190370.9	501.40	211.
Cycle	21.93	66.	.0005	14.88	.00	1.00	46.

## WEIGHTS

Name: **OWE**

Description: Operating Weight Empty. (lb.)

Location: Output file

Details: Detailed weight statement (see below)

Name: **Wpayload**

Description: Payload weight. (lb.)

Location: Output file  
 Details: Detailed weight statement (see below)

Name: **Wfuel**

Description: Fuel weight (lb.)

Location: Output file

Details: Detailed weight statement (see below)

The weight information for the aircraft file can be extracted from the weight statement given in the output of ACSYNT. Figure A-4-2 is an example of that section, including the shaded lines where the pertinent information can be extracted.

*Figure A-4-2. Weight Statement Example*

BOEING 757-200						
Qmax: 700.						
Design Load Factor: 2.78						
Ultimate Load Factor: 3.50						
Structure and Material: Aluminum Skin, Stringer						
Wing Equation: Sanders Equation						
Body Equation: Sanders Equation						
Component	Pounds	Kilograms	Percent	Slope	Tech	Fixed
Airframe Structure	72650	32954	29.09			No
Wing	24458	11094	9.79	1.00	1.00	No
Fuselage	27632	12534	11.06	1.00	1.00	No
Horizontal Tail (Low)	4599	2086	1.84	1.00	1.00	No
Vertical Tail	2215	1005	.89	1.00	1.00	No
Nacelles	3979	1805	1.59	1.00	1.00	No
Landing Gear	9767	4430	3.91	1.00	1.00	No
Propulsion	21742	9862	8.70			No
Engines (2)	20070	9104	8.04	1.00	1.00	No
Fuel System	1672	759	.67	1.00	1.00	No
Fixed Equipment	33917	15385	13.58		1.00	No
Hyd & Pneumatic	1499	680	.60	1.00		No
Electrical	2855	1295	1.14	1.00		No



Figure A-4-2. Weight Statement Example (Continued)

Component	Pounds	Kilograms	Percent	Slope	Tech	Fixed
Avionics	1594	723	.64	1.00		No
Instrumentation	869	394	.35	1.00		No
De-ice & Air Cond	3721	1688	1.49	1.00		No
Aux Power System	631	286	.25	1.00		No
Furnish & Eqpt	19388	8794	7.76	1.00		No
Seats & Lavatories	10516	4770	4.21	1.00		No
Galley	3346	1518	1.34	1.00		No
Misc Cockpit	234	106	.09	1.00		No
Cabin Finishing	3639	1651	1.46	1.00		No
Cabin Emergency	1291	585	.52	1.00		No
Equip						
Cargo Handling	362	164	.14	1.00		No
Flight Controls	3360	1524	1.35	1.00		No
Empty Weight	128309	58201	51.37			
Operating Items	10609	4812	4.25			No
Flight Crew (2)	340	154	.14			No
Crew Baggage & Provisions	300	136	.12			No
Flight Attendants (6)	780	354	.31			No
Unusable Fuel and Oil	600	272	.24			No
Passenger Service	8589	3896	3.44			No
Cargo Containers	0	0	.00			No
Operating Weight Empty	138918	63013	55.62			
Fuel	63055	28602	25.24			
Payload	47800	21682	19.14			No
Passengers	40630	18430	16.27			No
Baggage	7170	3252	2.87			No
Cargo	0	0	.00			No
Calculated Weight	249773	113297	100.00			No

## GEOMETRIC PARAMETERS

Figure A-4-3 is an example of the summary output from ACSYNT. It shows a detail section for the geometric parameters of the vehicle being modeled in ACSYNT. The shaded section shows where the wing area can be found. Remember to convert the ACSYNT wing area (given in ft<sup>2</sup>) to m<sup>2</sup>

Name: **swing**

Description: Wing reference area (ft<sup>2</sup>.)

Location: Output file

Details: Summary output (see below)

*Figure A-4-3. Summary—ACSYNT Output*

General		Fuselage			Wing	HTail	VTail
<b>WG</b>	<b>249773.</b>	<b>Length</b>	<b>155.0</b>	<b>Area</b>	<b>1950.0</b>	<b>680.0</b>	<b>298.0</b>
W/S	128.1	Diameter	12.0	Wetted Area	3377.3	1366.1	597.8
T/W	.35	Volume	16130.2	Span	118.2	58.3	19.6
N(Z) Ult	3.5	Wetted Area	5581.4	L.E. Sweep	28.5	29.9	42.3
Crew	2.	Fineness Ratio	12.9	C/4 Sweep	25.0	25.3	35.0
Passeng	239			Aspect Ratio	7.16	5.00	1.29
Engines		Weights		Taper Ratio	.30	.33	.30
Number	2	W	WG	T/C Root	.12	.12	.10
Length	13.8	Struct.	72650	29.1	T/C Tip	.09	.07
Diam.	6.9	Propul.	21742	8.7	Root Chord	25.5	17.6
Weight	8362.3	Fix. Eq.	33917	13.6	Tip Chord	7.5	5.7
TSLS	43200.	Fuel	63655	25.5	M.A. Chord	18.1	12.7
SFCSLS	.27	Payload	47800	19.1	Loc. Of L.E.	53.6	149.1
		Oper It	10609	4.2			126.9

## MISCELLANEOUS

Name: **actype**

Description: Aircraft type (1-Jet / 2 - Turboprop)

Location: Input file

Details: Propulsion module input section. If the engine analysis is using cycle mode, then the engine type will be specified as shown below in the input section. If the engine type is '7' then the aircraft is using propellor engine model. Otherwise, all other engine modes are turbofan models.

\*\*\*\* BOEING 757-200 / PW 2037 TURBOFAN \*\*\*\*

5 **{Engine Type}**

\$LEWIS

AENTW = 0.0,

AENWT = 0, AENLE = 0.0, AENDIA = 0,

## AERODYNAMICS

Name: **CDO**

Description: Zero-lift drag coefficient

Location: Output file

Details: Aerodynamics output section.

Name: **CD2**

Description: Induced drag factor.

Location: Output file

Details: Aerodynamics output section.

Figure A-4-4 is an example of the aerodynamic output section from ACSYNT. Use this section to obtain the zero-lift drag ( $C_{D0}$ ) and the induced drag factor. The shaded lines show where this information is located in the output. Be aware that you might want to compare these factors over several Mach numbers and altitudes before selecting the values, since the Mission Generator uses a fixed drag polar over all flight regions.

Figure A-4-4. Aerodynamic Output from ACSYNT

Mach	=	.77	C.G. Location		=	71.3 ft, .25 cbar					
Altitude	=	36000.	Reynolds Number per foot		=	1793352.					
Parasite Drag		Induced Drag									
Friction	.0129	Alpha	C1	Cd	L/D	e	Zone	Cm	Cdtrim	Deltrim	StMrg
Body	.0056	.0	.000	.0159	.0	.00	2	.000	.0000	.0	.767
Wing	.0042	1.2	.144	.0172	8.3	.67	2	.000	.0002	-1.1	.767
Strakes	.0000	2.4	.284	.0212	13.4	.67	2	.000	.0009	-2.2	.767
H. Tail	.0022	3.6	.420	.0276	15.2	.67	2	.000	.0019	-3.2	.766
V. Tail	.0009	4.8	.554	.0363	15.3	.67	2	.000	.0032	-4.2	.765
Canard	.0000	6.0	.684	.0472	14.5	.66	2	.000	.0049	-5.2	.763
		7.1	.812	.0601	13.5	.66	2	.000	.0068	-6.2	.761
		8.3	.936	.0749	12.5	.66	2	.000	.0090	-7.2	.759
		9.5	1.05	.0915	11.6	.66	2	.000	.0114	-8.1	.757
			7								
		10.7	1.17	.1099	10.7	.65	2	.000	.0140	-9.0	.754
			6								
Interference	.0029										
Wave	.0000										
External	.0000	Slope Factors									
Tanks	.0000	C1/Alpha (per radian)								6.3225	
Bombs	.0000	Cd1/C1^2								.0679	
Stores	.0000	Alpha Transition Zone 2-3								10.637	
Extra	.0000										
Camber	.0000	Programmed Flap Setting								0.	
Cdmin	.0159	Flap Type		Single		278. sq. ft					

Name: **CD16**

Description: Transonic drag rise factor.

Location: Derived

Details: Aerodynamics output section. This variable is part of the equation for the drag polar. The drag coefficient is given by,

$$C_D = (C_{D0} + C_{D2} C_L^2) \cdot (1.0 + C_{D16} M^{16})$$

To derive the CD16 value from ACSYNT output will require several flight points from the aerodynamics and then a curve-fit from the data into the form shown above.

The transonic drag rise factor is used only when the Mach number approaches 1.0 (evidence from the large exponent (16) on the Mach number)

Figure A-4-5 is a portion of the aerodynamics graph file (AERO\_DATA) showing the list of Mach numbers and drag coefficients.

The default value for most transonic aircraft found in the BADA library is CD16=2.

Figure A-4-5. Aero Graph File

{Mach}	{Cdmin}										
.600	.000	.0029	.0000	.0000	.0133	.0000	.0000	.0162	.0000	.0000	36000.00
.650	1.189	.0029	.0000	.0000	.0132	.0000	.0000	.0161	.0000	.0000	36000.00
.700	2.375	.0029	.0000	.0000	.0131	.0000	.0000	.0160	.0000	.0000	36000.00
.750	3.558	.0029	.0000	.0000	.0130	.0000	.0000	.0159	.0000	.0000	36000.00
.775	4.739	.0029	.0000	.0000	.0129	.0000	.0000	.0159	.0000	.0000	36000.00
.800	5.916	.0027	.0013	.0000	.0129	.0000	.0000	.0169	.0000	.0000	36000.00
.825	7.091	.0023	.0090	.0000	.0128	.0000	.0000	.0241	.0000	.0000	36000.00
.850	8.263	.0019	.0215	.0000	.0128	.0000	.0000	.0362	.0000	.0000	36000.00
.200	9.432	.0029	.0000	.0000	.0138	.0000	.0000	.0167	.0000	.0000	.00
.150	12.258	.0029	.0000	.0000	.0138	.0000	.0000	.0167	.0000	.0000	.00

## PROPULSION: FUEL FLOWS

The jet propulsion fuel-flow variables, Cf1 and Cf2, are used in a linear model for fuel flow as a function of cruise velocity. Figure A-4-6 is an example of the propulsion output section.

This model takes the form

$$\eta = C_{f1} \left( 1 + \frac{V_{tas}}{C_{f2}} \right)$$

---

and produces flows of kg/min/kN (Caution: ACSYNT sfc's are produced in lb/hr/lb so be sure to use the proper conversions where illustrated

Name: **Cf1**

Description: Fuel flow coefficient

Location: DERIVED / Output file

Details: Propulsion output section.

Name: **Cf2**

Description: Fuel flow coefficient (Velocity Correction)

Location: DERIVED / Output file

Details: Propulsion output section.

Name: **Cf3**

Description: Fuel flow coefficient for descent

Location: DERIVED / Output file

Details: Propulsion output section.

Name: **Cf4**

Description: Fuel flow coefficient (descent Velocity Correction)

Location: DERIVED / Output file

Details: Propulsion output section.

Figure A-4-6. Propulsion Output Section

ENGINE DIAMETER = 6.85 FEET ENGINE LENGTH = 13.84 FEET ENGINE WEIGHT = 8362.34 POUNDS BYPASS RATIO = 4.40 NO OF ENGINES = 2. DRAG REF AREA = 1950.01 SQ FEET PWCC = PERCENT OF ENGINE CORRECTED AIRFLOW THRUST = ENGINE THRUST (POUNDS PER ENGINE) SFC = ENGINE SPECIFIC FUEL CONSUMPTION THRUSTU = THRUST PER ENGINE IN LBS, W/O INSTAL DRAG CORR SFCU = SFC, 1/HR, W/O INSTALLATION DRAG CORR CDINS = TOT INSTALLATION DRAG COEF PER A/C (SWING REF)							
MACH	ALT	PWCC	THRUST	THRUSTU	SFC	SFCU	CDINS
.550	36000.	100.	7690.	7752.	.381	.378	.0006
		98.	6861.	6924.	.376	.372	.0006
		96.	6050.	6113.	.372	.368	.0006
		94.	5257.	5320.	.371	.367	.0006
		92.	4482.	4543.	.374	.369	.0006
		90.	3722.	3781.	.383	.377	.0006
		88.	2977.	3032.	.403	.395	.0006
		86.	2234.	2285.	.444	.434	.0005
		84.	1473.	1518.	.547	.531	.0005
		82.	954.	1001.	.670	.638	.0005
		80.	427.	479.	1.146	1.022	.0005
.600	36000.	100.	7761.	7824.	.396	.393	.0005
		98.	6917.	6980.	.391	.387	.0005
		96.	6091.	6156.	.387	.383	.0006
		94.	5287.	5351.	.387	.382	.0005
		92.	4502.	4565.	.390	.385	.0005
		90.	3735.	3796.	.400	.394	.0005
		88.	2987.	3045.	.421	.413	.0005
		86.	2248.	2301.	.463	.452	.0005
		84.	1502.	1549.	.563	.546	.0004
		82.	980.	1031.	.684	.650	.0004
		80.	454.	510.	1.130	1.006	.0005
.650	36000.	100.	7871.	7935.	.411	.407	.0005
		98.	7008.	7073.	.405	.402	.0005
		96.	6166.	6232.	.402	.398	.0005
		94.	5346.	5412.	.402	.397	.0005

Figure A-4-6. Propulsion Output Section (Continued)

MACH	ALT	PWCC	THRUST	THRUSTU	SFC	SFCU	CDINS
		92.	4549.	4614.	.406	.401	.0005
		90.	3771.	3835.	.417	.410	.0005
		88.	3018.	3078.	.438	.429	.0004
		86.	2278.	2334.	.480	.469	.0004
		84.	1539.	1589.	.578	.560	.0004
		82.	1014.	1068.	.696	.661	.0004
		80.	486.	546.	1.112	.990	.0004
.700	36000.	100.	8020.	8086.	.425	.422	.0004
		98.	7135.	7202.	.420	.416	.0004
		96.	6273.	6340.	.417	.413	.0004
		94.	5434.	5502.	.417	.412	.0004
		92.	4622.	4688.	.422	.416	.0004
		90.	3831.	3896.	.433	.426	.0004
		88.	3068.	3130.	.455	.446	.0004
		86.	2323.	2381.	.497	.485	.0004
		84.	1587.	1640.	.592	.573	.0003

### Steps for creating a fuel flow model from ACSYNT output.

1. Run the ACSYNT model with several propulsion output points with variations in Mach number at a selected altitude proper for the chosen aircraft.
2. The information will be in an output format as shown above. Select the part-power line appropriate for the cruise fuel-flow model. (For the Mission Generator, the fuel flow is computed at various points, primarily at Thrust=Drag. Since this can range over various power settings, it is recommended to select a value the aircraft designer believes best models flight conditions. Also, the designer can select various different part-power settings for different Mach numbers.) From this output, the designer should create an array of data for sfc and velocities.
3. Convert the Mach numbers to flight velocities using the local speed of sound.
4. Convert the sfc's from lb/hr/lb to kg/min/kN by using the following conversion string  $(.4535\text{kg/lb}) * (1\text{hr}/60\text{min}) * (1\text{lb}/4.448\text{N}) * (1000\text{N}/1\text{kN})$
5. Since we have a linear model, we will need to estimate the slope and the intercept.



First, compute the slope from the following equation:

$$slope = \frac{C_{f1}}{C_{f2}} = \frac{(sfc_{i+1} - sfc_i)}{(V_{i+1} - V_i)}$$

If the data from the ACSYNT model is not close to linear, an estimated validity zone will have to be selected, and the data fit in that region only.

6. Next, compute the intercept from the data by using the slope and a data point from the array:

$$Intercept = C_{f1} = sfc_i - V_i \frac{(sfc_{i+1} - sfc_i)}{(V_{i+1} - V_i)}$$

7. Now that Cf1 has been determined, use the computed slope to determine Cf2,

$$C_{f2} = C_{f1} / slope = \frac{C_{f1}}{\frac{sfc_{i+1} - sfc_i}{V_{i+1} - V_i}}$$

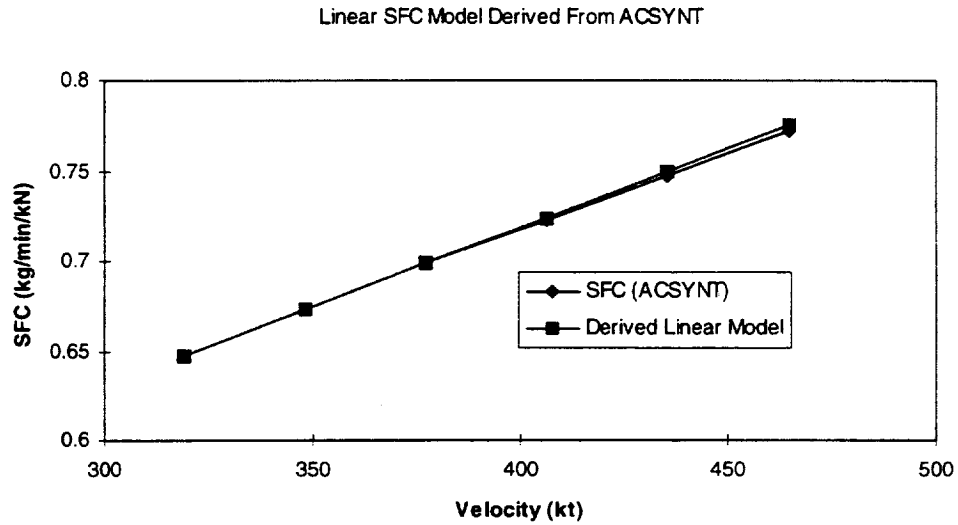
An example is shown in Table A-4-2:

*Table A-4-2. Sample Computation of Velocity Correction Coefficient Cf2*

Mach	Velocity	Altitude	PWCC	Thrust ( )	Thrust (U)	lb/hr/lb	kg/min/kN	Model	Slope	Intercept
0.55	319.3792	36000	100	7690	7752	0.381	0.647563	0.647415	0.000878	0.367122
0.6	348.4137	36000	100	7761	7824	0.396	0.673058	0.672907	0.000878	0.367122
0.65	377.4482	36000	100	7871	7935	0.411	0.698552	0.698399	0.00082	0.389218
0.7	406.4826	36000	100	8020	8086	0.425	0.722347	0.723892	0.000878	0.365423
0.75	435.5171	36000	100	8209	8275	0.44	0.747842	0.749384	0.00082	0.390717
0.8	464.5516	36000	100	8430	8505	0.454	0.771637	0.774876		

The computed coefficients for this model are Cf1 = .367 and Cf2  $\frac{.367}{.878} = 417.9$ .

Figure A-4-7. Trajectory Output



The descent fuel flow coefficients can be determined in the same fashion, using the descent power settings.

## PROPULSION: THRUST MODEL

Thrust for jet engines are given as a quadratic function of altitudes. The equation has three coefficients for altitude correction:

$$T_{\max \text{ climb}} = C_{t1} (1 - h / C_{t2} + C_{t3} h^2)$$

Name: **Ct1**

Description: Thrust coefficient (N)

Location: DERIVED / Output file

Details: Propulsion output section.

Name: **Ct2**

Description: Thrust coefficient (First-order altitude correction) (ft.)

Location: DERIVED / Output file

Details: Propulsion output section.

Name: **Ct3**

Description: Thrust coefficient (Second order altitude correction) (1/ft<sup>2</sup>)

Location: DERIVED / Output file

Details: Propulsion output section.

## Steps for Creating a Thrust Model From ACSYNT Output.

The thrust model is a quadratic function of the altitude. The Mission Generator aircraft file assumes that the thrust model is for the whole aircraft, so thrust values are the total components of all engine output. With this in mind, as the ACSYNT output is translated, be sure to multiply the thrust from ACSYNT by the number of engines on the vehicle.

1. Run the ACSYNT model with several propulsion output points with variations in altitude at a selected Mach number proper for the chosen aircraft.
2. As in the case when constructing the fuel flow model, collect the data as an array of thrust and altitude points for the selected max power settings, converting from lbs to Newtons.
3. Ct1 will be the reference thrust for sea level. This value should be shown at the end of the ACSYNT propulsion output. This value should be converted to Newtons, and should be multiplied by the number of engines on the aircraft.

$$T_{\max climb}(0) = C_{t1}$$

---

Example section of output file:

80.	603.	681.	1.070	.947	.0004
78.	-20.	68.	-23.161	6.903	.0004

SEA-LEVEL STATIC THRUST = 43200. (MAX)

SEA-LEVEL SFC = .270

4. In order to obtain Ct2 and Ct3, we will have to use two data points and solve two equations.

First, we can rewrite the thrust equation, multiplying out the Ct1 and renaming the coefficients

$$T(h1) = (C1 - C2h1 + C3h1^2)$$

$$T(h2) = (C1 - C2h2 + C3h2^2)$$

We can rewrite C2 as a function of C3 from equation 1, and then solve equation 2 for C3.

$$C2 = \left[ \frac{C1 - T(h1) + C3h1^2}{h1} \right]$$

$$C3 = \frac{\left[ C1 \left\{ 1 - \frac{h2}{h1} \right\} + T(h1) \frac{h2}{h1} - T(h2) \right]}{(h1^2 h2 - h1^2)}$$

Once C1, C2, and C3 are obtained, we can then compute Ct1, Ct2 and Ct3 from the following:

$$C1 = Ct1$$

$$C2 = Ct1/Ct2$$

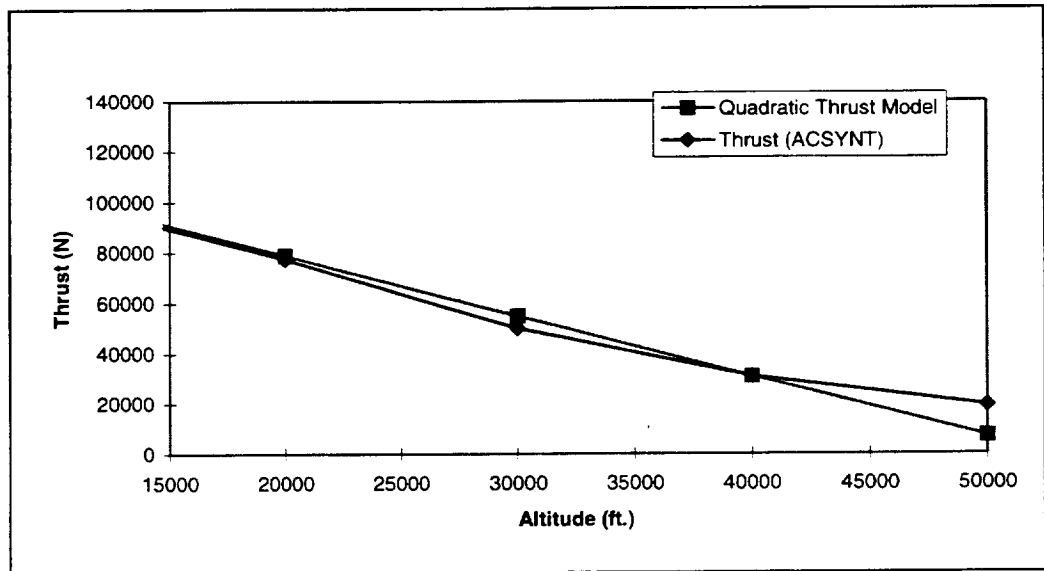
$$C3 = Ct1 * Ct3$$

An example spreadsheet is shown below with a portion of the ACSYNT propulsion output file and the actual computed coefficients for this model.

Table A-4-3. Sample Thrust Coefficients

Computed Coefficients		Altitude	Model	Thrust (lb)	T (N)
C1	126768	50000	7000	4312	19180
C2	2.395353	40000	30954	6959	30954
C3	-1.5E-10	30000	54907	11244	50013
		20000	78861	17401	77398
Ct1	126768	10000	102814	22825	101526
Ct2	52922.47	0	126768	28500	126768
Ct3	-1.2E-15				

Figure 8. Thrust Curve Model Constructed from ACSYNT Data



## Obtaining Fuel-Burn Rates from ACSYNT Output (Trajectory)

The fuel burn rates for the aircraft input file can be derived from the ACSYNT output in several ways. The easiest method is to examine the trajectory output section, and for each phase of the ACSYNT trajectory, simply compute a fuel burn rate per hour. This will not be as accurate as using propulsion information, however, a rough estimate can be obtained quickly. Be careful of phases that cross too much altitude, as the result may be averaged over too much space or time.

For example, a portion of the trajectory output from ACSYNT is shown in Figure A-4-9. For the example climb segment, we can compute the fuel burn rate (in lb/hr) as:

$$(1070.2 \text{ lb} / 2.95 \text{ min}) * (60 \text{ min/hr}) = 21,762 \text{ lb/hr for climb.}$$

This is a value for the whole aircraft, and is a simple approximation based on the operating conditions reported in the trajectory output.

*Figure A-4-9. Trajectory Output*

Mission 1 (PAYLOAD = 47800. LB)							
PHASE	M	H	CL	ALPHA	WFUEL	TIME	VEL
SFC(I)	THRUST(I)	CD	GAMMA	W	WA	Q	
SFC(U)	THRUST(U)	CDINST	L/D	THR/THA	PR	X	
WARM-UP		0.			366.1	5.80	
1.02	3703.						
TAKEOFF	.16	0.	3.4486	14.53	784.1	2.00	176.
	.30	78054.	.8516	3.95	249415.0	1357.72	37.
	.30	78054.	.0024	4.05	1.00	1.00	3137.
2 <sup>ND</sup> SEG	.16	400.	3.4486	14.53			176.
	.30	39027.	.8516	5.05	249415.0	1357.72	37.
	.30	78054.	.0024	4.05	1.00	1.00	
CLIMB	.46	10000.	.6931	7.12	1070.2	2.95	491.
	.41	47689.	.0514	6.05	247560.8	997.69	212.
Cycle	.40	47984.	.0007	13.49	.96	1.00	13.
CRUISE	.80	36000.	.5421	4.56	23559.2	196.00	775.
	.45	15103.	.0363	.00	224001.5	452.77	213.
Cycle	.45	15256.	.0004	14.94	.72	1.00	1500.
CRUISE	.80	38000.	.5075	4.26	34272.8	326.90	774.
	.45	12811.	.0339	.00	189728.8	407.15	194.
Cycle	.44	12951.	.0004	14.98	.67	1.00	2500.

# Converting FLOPS Output to Mission Generator Aircraft Format

Parameters listed in the aircraft file—an input file to the Mission Generator—are taken from BADA (Base of Aircraft Data) coefficients. (See “Aircraft Data File” documentation.) These BADA coefficients, in turn, are related to data in a FLOPS (Aircraft Synthesis) file. Described below is the relationship between FLOPS data and BADA coefficients, which could be used to create a mission generator aircraft file from a FLOPS data file. Only the relationships needed to describe the aircraft file parameters are given. See the aircraft file documentation for a list of the parameters and brief descriptions.

## PERFORMANCE AND VEHICLE LIMITS

Performance limits are necessary for the Mission Generator to apply meaningful bounds to the vehicle parameters in the optimization problem. Typically, an aircraft designer will have some estimate of maximum performance limits for a given configuration. These values should be used in the aircraft file when available. Performance limit estimation algorithms are available in most aircraft design programs.

FLOPS contains routines that compute operating envelop information. Specifically, excess power and sustained g's are computed. The user can obtain a rough estimate for the vehicle limits from the output of the excess power plots. The user will need to fully examine the estimated limits for the aircraft concept and determine from the envelop the meaningful values for the following aircraft inputs to the Mission Generator. If the limits are not easily obtained, estimated values should bound the performance of the vehicle in such a way that the optimization algorithm has ‘room’ to find a satisfactory fuel minima.

Name:	<b>Vmo</b>
Description:	Maximum operating velocity (kt)
Location:	Derived
Details:	Using the Mmo value found in the excess power plot file, derive Vmo from Mmo and the selected maximum altitude for the velocity.

---

Name: **Mmo**

Description: Maximum operating Mach number

Location: Input file

Details: Found in the excess power output plot file.

Example: The following is an example of the excess power plot file showing maximum Machs at various altitudes:

```
TITLE = "FLOPS GENERATED OUTPUT"
VARIABLES = MACH, ALT, NZ, PS
##
## ALTITUDE = 40000. 60000. 2000.
## MACH = .600 .950 .020
##
##
## PS CONTOUR PLOT DATA FOR NZ = .00
## WEIGHT = 256823.2
## STORE TABLE = 0
##
ZONE T="PS, NZ= .00", I=18, J=11, F=POINT
```

##MACH	ALTITUDE	NZ	PS
6000	40000.	.000	745.0
.8000	54000.	.000	-296.0
.8200	54000.	.000	-210.1
.8400	54000.	.000	-123.4
.8600	54000.	.000	-42.7
.8800	54000.	.000	17.9
.9000	54000.	.000	29.5
.9200	54000.	.000	-38.8



Name: **Hmo**

Description: Maximum operating altitude (ft)

Location: Output file

Details: The value for the maximum altitude for 1G, steady, level flight can be computed using the excess power routine in FLOPS or the user can enter the design altitude limit.

Name: **Vstall**

Description: Cruise stall velocity (kt)

Location: Derived from Mstall

Details: This information can come from user-specified configuration data, or the user can estimate from the output of takeoff routines and scale the velocities to the appropriate cruise level.

Name: **Vave**

Description: Cruise velocity at average mass cruise point (kt)

Location: Output file

Details: Trajectory output provides data for the cruise velocity.

Name: **Mave**

Description: Cruise Mach at average mass cruise point

Location: Output file

Details: Trajectory output provides data for the cruise Mach.

---

Name: **Have**  
Description: Average cruise altitude. (ft.)  
Location: Output file  
Details: Trajectory output provides data for the average cruise altitude.

Name: **Hstart**  
Description: Cruise starting altitude. (ft.)  
Location: USER DEFINED / Output file  
Details: This is the altitude the user wishes the airplane to start the entire cruise phase. The selected altitude should be chosen to reflect where the climb segment ends and the cruise segment begins. An alternative to user input is selecting the end of the climb phase in the FLOPS trajectory output file. This variable should default to 23,000 ft. for jet, and 10,000 ft. for the turboprop.

Name: **Hend**  
Description: Cruise ending altitude. (ft.)  
Location: USER DEFINED / Output file  
Details: This is the altitude the user wishes the airplane to end the entire cruise phase. The selected altitude should be chosen to reflect where the cruise ends and the descent segment begins. An alternative to user input is selecting the altitude at the beginning of the descent phase in the FLOPS trajectory output file. Usually, this variable defaults to 10,000 ft.

Name: **Mend**  
Description: Cruise ending Mach

Location:      USER DEFINED / Output file

Details:        This is the Mach number the user specifies for which the airplane ends the entire cruise phase. This Mach should be chosen to reflect entry speed into the descent and landing phase. Typically this is chosen to be 0.392 to reflect the 250 kt speed limit at and below 10,000 ft in the terminal operating area. An alternative to user input is selecting the Mach at the beginning of the descent phase in the FLOPS trajectory output file.

Table A-4-4 is an example section from mission results in FLOPS Output.

*Table A-4-4. Summary Data For Cruise Schedule 1*

Weight	Altitude	Thrust Available   Required		Fuel Flow	SFC	Mach Number	Velocity	Specific range	Rate of climb	CL	L/D	Engines operating
177000	37761	15984	15603.	9658.	.6190	.8000	458.9	.04751	100.0	.2550	11.34	2.0
180000	37445.	16230.	15842.	9807.	.6191	.8000	458.9	.04679	100.0	.2554	11.36	2.0
183000	37127.	16476.	16082.	9957.	.6191	.8000	458.9	.04608	100.0	.2557	11.38	2.0

## WEIGHTS

Name:            **OWE**

Description:    Operating Weight Empty. (lb.)

Location:        Output file

Details:         Detailed weight statement (see below)

Name:            **Wpayload**

Description:    Payload weight. (lb.)

Location:        Output file

Details:         Detailed weight statement (see below)

Name: **Wfuel**

Description: Fuel weight (lb.)

Location: Output file

Details: Detailed weight statement (see below)

The weight information for the aircraft file can be extracted from the weight statement given in the output of FLOPS. Figure A-4-10 is an example of that section, including the shaded lines where the pertinent information can be extracted.

*Figure A-4-10. Weight Statement*

Mass and balance summary	Percent WREF	Pounds	Percent LREF	Hori CG- Inches
Wing	9.41	24154.	.0	.0
Horizontal Tail	2.31	5936.	.0	.0
Vertical Tail	.81	2085.	.0	.0
Fuselage	12.09	31048.	.0	.0
Landing Gear	3.79	9725.	.0	.0
Nacelle (Air Induction)	1.30	3332.	.0	.0
Structure Total	(29.70)	(76280.)	(.0)	(.0)
Engines	7.12	18292	.0	.0
Thrust Reversers	.00	0.	.0	92.8
Miscellaneous Systems	.31	784.	.0	.0
Fuel System-Tanks and Plumbing	.69	1764.	.0	.0
Propulsion Total	(8.11)	(20841.)	(.0)	(.0)
Surface Controls	1.43	3673.	.0	.0
Auxiliary Power	.47	1200.	.0	.0
Instruments	.28	722.	.0	.0
Hydraulics	.75	1933.	.0	.0
Electrical	.91	2343.	.0	.0
Avionics	.84	2151.	.0	.0
Furnishing and Equipment	7.91	20322.	.0	.0
Air Conditioning	.89	2292.	.0	.0
Anti-icing	.09	226.	.0	.0
System and Equipment Total	(13.57)	(34861.)	(.0)	(.0)
Weight Empty	51.39	131981.		
Crew and Baggage Flight, 3	.26	675.	.0	.0
Cabin, 7	.44	1130.	.0	.0

Figure A-4-10. Weight Statement (Continued)

Mass and balance summary	Percent WREF	Pounds	Percent LREF	Hori CG-Inches
Unusable Fuel	.31	807.	.0	.0
Engine Oil	.07	183.	.0	.0
Passenger Service	1.33	3415.	.0	.0
Cargo Containers	.61	1575.	.0	.0
<b>Operating Weight</b>	<b>54.42</b>	<b>139766.</b>	<b>.0</b>	<b>.0</b>
<b>Passengers, 200</b>	<b>12.85</b>	<b>33000.</b>	<b>.0</b>	<b>.0</b>
<b>Passenger Baggage</b>	<b>3.12</b>	<b>8000.</b>	<b>.0</b>	<b>.0</b>
Cargo	.00	0.	.0	.0
Zero Fuel Weight	70.39	180766.	.0	.0
<b>Mission Fuel</b>	<b>29.61</b>	<b>76054.</b>	<b>.0</b>	<b>.0</b>
Ramp (Gross) Weight	100.00	256820.	.0	.0

## GEOMETRIC PARAMETERS

Figure A-4-11 is an example of the summary output from FLOPS. It shows a detail section for the geometric parameters of the vehicle being modeled in FLOPS. The shaded section shows where the wing area can be found. Remember to convert the FLOPS wing area (given in ft<sup>2</sup>) to m<sup>2</sup>

Name: **swing**

Description: Wing reference area (ft<sup>2</sup>.)

Location: Output file

Details: Summary output (see below)

Figure A-4-11. Aerodynamic Data

MACH NUMBER		ALTITUDE		REFERENCE AREA		TECHNOLOGY LEVEL	
.80		40000FT		3541.00 SQ FT		1.50	
COMPONENT	SWET SQ FT	LENGTH FT	FINENESS RATIO	FORM FACTOR	RN MILLIONS	CF	CDF
WING	5582.77	32.19	.1100	1.425	49.3	.00218	.00491
HORIZONTAL TAIL	2004.97	26.55	.1100	1.425	40.7	.00225	.00181
VERTICAL TAIL	1050.80	24.54	.1100	1.425	37.6	.00227	.00096
FUSELAGE	5828.25	151.43	8.9709	1.134	232.0	.00176	.00328
NACELLE	382.77	15.47	1.7500	2.798	23.7	.00244	.00074
NACELLE	382.77	15.47	1.7500	2.798	23.7	.00244	.00074
MISCELLANEOUS						.00075	
TOTAL	15232.32					.00306	.01318

## MISCELLANEOUS

Name: **actype**

Description: Aircraft type (1-Jet / 2 - Turboprop)

Location: User-defined

Details: The user will have to specify aircraft type.

## AERODYNAMICS

Name: **CDO**

Description: Zero-lift drag coefficient

Location: Output file

Details: Aerodynamics output section.

Name: **CD2**

Description: Induced drag factor.

Location: Output file

Details: Aerodynamics output section.

Below is an example of the aerodynamic output section from FLOPS. Use this section to obtain the zero-lift drag ( $C_{D0}$ ) and the induced drag factor. The shaded lines show where this information is located in the output. Be aware that you might want to compare these factors over several Mach numbers and altitudes before selecting the values, since the Mission Generator uses a fixed drag polar over all flight regions.

Figure A-4-12. Aerodynamic Output From FLOPS

Drag Polars				Altitude = 40000. FT									
Mach	CL												
	.000	.050	.100	.150	.200	.250	.300	.350	.400	.450	.500	.550	.600
.200	.01736	.01776	.01883	.02025	.02224	.02548	.02925	.03465	.04222	.05098	.06138	.07616	.09456
.300	.01623	.01663	.01770	.01910	.02107	.02435	.02811	.03351	.04108	.04985	.06025	.07502	.09342
.400	.01544	.01584	.01691	.01831	.02027	.02357	.02733	.03273	.04029	.04906	.05946	.07424	.09263
.500	.01483	.01523	.01630	.01771	.01968	.02295	.02672	.03212	.03968	.04845	.05885	.07362	.09202
.600	.01431	.01471	.01578	.01721	.01921	.02243	.02620	.03160	.03916	.04793	.05833	.07311	.09150
.700	.01415	.01455	.01563	.01710	.01914	.02226	.02604	.03144	.03900	.04777	.05817	.07295	.09134
.800	.01426	.01468	.01579	.01736	.01949	.02235	.02616	.03157	.03913	.04789	.05828	.07306	.09144
.850	.01458	.01503	.01623	.01788	.02007	.02288	.02689	.03291	.04028	.04887	.05915	.07388	.09217
.875	.01509	.01556	.01683	.01859	.02094	.02401	.02857	.03486	.04226	.05100	.06144	.07604	.09405
.900	.01637	.01693	.01834	.02023	.02278	.02642	.03169	.03791	.04591	.05522	.06587	.08008	.09762
.925	.01887	.01954	.02125	.02328	.02603	.03055	.03676	.04384	.05175	.06151	.07315	.08710	.10436
.950	.04752	.04828	.05038	.05279	.05610	.06275	.07206	.07942	.08771	.09742	.10756	.12264	.14161

Figure A-4-12. Aerodynamic Output From FLOPS (Continued)

Design Mach Number = .877, Design CL = .292			
Mach	CDF	CDC	Buffet CL
.200	.01712	.00000	.619
.300	.01598	.00000	.604
.400	.01520	.00000	.577
.500	.01458	.00000	.537
.600	.01406	.00000	.490
.700	.01360	.00030	.448
.800	.01318	.00083	.394
.850	.01298	.00135	.358
.875	.01289	.00195	.337
.900	.01279	.00329	.312
.925	.01270	.00586	.284
.950	.01260	.03446	.254
CL	CDI		
.000	.00000		
.050	.00023		
.100	.00093		
.150	.00210		
.200	.00373		
.250	.00582		
.300	.00838		
.340	.01141		
.400	.01490		
.450	.01886		
.500	.02329		
.550	.02818		

Name: **CD16**

Description: Transonic drag rise factor.

Location: Derived

Details: Aerodynamics output section. This variable is part of the equation for the drag polar. The drag coefficient is given by,

$$C_D = (C_{D0} + C_{D2} C_L^2) \cdot (10 + C_{D16} M^{16})$$



To derive the  $C_{D16}$  value from FLOPS output will require several flight points from the aerodynamics and then a curve-fit from the data into the form shown above.

The transonic drag rise factor is used only when the Mach number approaches 1.0 (evidence from the large exponent (16) on the Mach number). Use the same section shown above that was used to build the zero-lift and induced drag variables to construct a fit for  $C_{D16}$ . The default value for most transonic aircraft found in the BADA library is  $C_{D16}=2$ .

## PROPULSION: FUEL FLOWS

The jet propulsion fuel-flow variables, Cf1 and Cf2, are used in a linear model for fuel flow as a function of cruise velocity.

This model takes the form

$$\eta = C_{f1} \left( 1 + \frac{V_{\text{cruise}}}{C_{f2}} \right)$$

and produces flows of kg/min/kN (Caution: FLOPS sfc's are produced in lb/hr/lb so be sure to use the proper conversions where illustrated)

Name:	<b>Cf1</b>
Description:	Fuel flow coefficient
Location:	DERIVED / Output file
Details:	Propulsion output section.

---

Name: **Cf2**  
Description: Fuel flow coefficient (Velocity Correction)  
Location: DERIVED / Output file  
Details: Propulsion output section.

Name: **Cf3**  
Description: Fuel flow coefficient for descent  
Location: DERIVED / Output file  
Details: Propulsion output section.

Name: **Cf4**  
Description: Fuel flow coefficient (descent Velocity Correction)  
Location: DERIVED / Output file  
Details: Propulsion output section.

Table A-4-5 is the format of a FLOPS propulsion output file

*Card Format: (F5.2, F10.0, F5.0, 3F10.0, 10X, 2F10.0)*

*Table A-4-5. Propulsion Format*

Columns	Description
1-5	Mach number
6-15	Altitude, feet
16-20	Power code
21-30	Gross thrust, lb.
31-40	Ram drag, lb.
41-50	Fuel flow, lb./hr.
61-70	Nox Emissions Index, grams/kilogram fuel (or Nox Emissions, lb./hr. or other engine parameter selected by user)

Figure A-4-13 is a sample portion of propulsion file.

*Figure A-4-13. Propulsion Output Section*

MACH	ALT	PC	THRUST	Ram Drag Fuel Flow	SFC	Nox Em
.80	40000	1	7122.1	4399.4	.61772	5.794
		2	6399.3	3937.1	.62106	5.012
		3	5586.1	3507.4	.62787	4.327
		4	5147.4	3264.0	.63411	3.955
		5	4715.1	3031.6	.64295	3.612
		6	4300.4	2812.3	.65396	3.297
		7	3883.4	2596.8	.66869	3.000
		8	3441.2	2376.9	.69070	2.713
		9	3027.5	2173.2	.71782	2.451
		10	2633.4	1983.4	.75314	2.212
		11	2264.8	1807.6	.79813	1.995
		12	1880.9	1638.1	.87092	1.794
		13	1528.2	1482.6	.97018	1.612
		14	1204.5	1339.9	1.11243	1.447
		15	927.9	1211.7	1.30587	1.298
		16	.0	781.5	.00000	.000

Figure A-4-13. Propulsion Output Section (Continued)

MACH	ALT	PC	THRUST	Ram Drag Fuel Flow	SFC	Nox Em
.80	35000	1	9064.5	5610.1	.61891	6.620
		2	8068.2	5020.6	.62226	.000
		3	7109.6	4472.6	.62909	.000
		4	6551.2	4162.2	.63534	.000
		5	6001.0	3865.9	.64420	.000
		6	5473.3	3586.2	.65523	.000
		7	4942.5	3311.4	.66999	.000
		8	4379.7	3031.0	.69204	.000
		9	3853.2	2771.2	.71921	.000
		10	3351.7	2529.2	.75460	.000
		11	2882.5	2305.1	.79968	.000
		12	2393.8	2088.8	.87260	.000
		13	1945.0	1890.6	.97206	.000
		14	1533.0	1708.6	1.11458	.000
		15	1180.9	1545.1	1.30840	.000
		16	.0	996.6	.00000	.000
.80	30000	1	11473.4	7268.4	.63350	8.777
		2	10212.4	6504.6	.63693	.000
		3	8999.0	5794.6	.64391	.000
		4	82.92.2	5392.5	.65031	.000
		5	7595.8	5008.5	.65938	.000
		6	6927.8	4646.3	.67067	.000
		7	6256.0	4290.2	.68578	.000
		8	5543.7	3926.9	.70835	.000
		9	4877.2	3590.4	.73617	.000
		10	4242.4	3276.7	.77238	.000
		11	3648.5	2986.4	.81853	.000
		12	3030.0	2706.3	.89317	.000
		13	2461.8	2449.5	.99497	.000
		14	1940.4	2213.7	1.14086	.000
		15	1494.7	2001.8	1.33924	.000

## Steps for Creating a Fuel Flow Model From FLOPS Output.

1. Run the FLOPS model with several propulsion output points.  
(Set IPLTTH=2 in the \$OPTIONS namelist)
2. The information will be in an output format as shown above. Select the part-power line appropriate for the cruise fuel-flow model. (For the Mission Generator, the fuel flow is computed at various points, primarily at Thrust=Drag. Since this can range over various power settings, it is recommended to select a value the aircraft designer believes best models flight conditions. Also, the designer can select various different part-power settings for different Mach numbers.) From this output, the designer should create an array of data for sfc and velocities.
3. Convert the Mach numbers to flight velocities using the local speed of sound.
4. Convert the sfc's from lb/hr/lb to kg/min/kN by using the following conversion string  
$$(.4535\text{kg/lb}) * (1\text{hr}/60\text{min}) * (11\text{bf}/4.448\text{N}) * (1000\text{N}/1\text{kN})$$
5. Since we have a linear model, we will need to estimate the slope and the intercept. First, compute the slope from the following equation:

$$\text{slope} = \frac{C_{f1}}{C_{f2}} = \frac{(sfc_{i+1} - sfc_i)}{(V_{i+1} - V_i)}$$

If the data from the FLOPS model is not close to linear, an estimated validity zone will have to be selected, and the data fit in that region only.

1. Next, compute the intercept from the data by using the slope and a data point from the array:

$$\text{Intercept} = C_{f1} = sfc_i - V_i \frac{(sfc_{i+1} - sfc_i)}{(V_{i+1} - V_i)}$$

2. Now that Cf1 has been determined, use the computed slope to determine Cf2,

$$C_{f2} = C_{f1} / \text{slope} = \frac{C_{f1}}{\left[ \frac{sfc_{i+1} - sfc_i}{V_{i+1} - V_i} \right]}$$

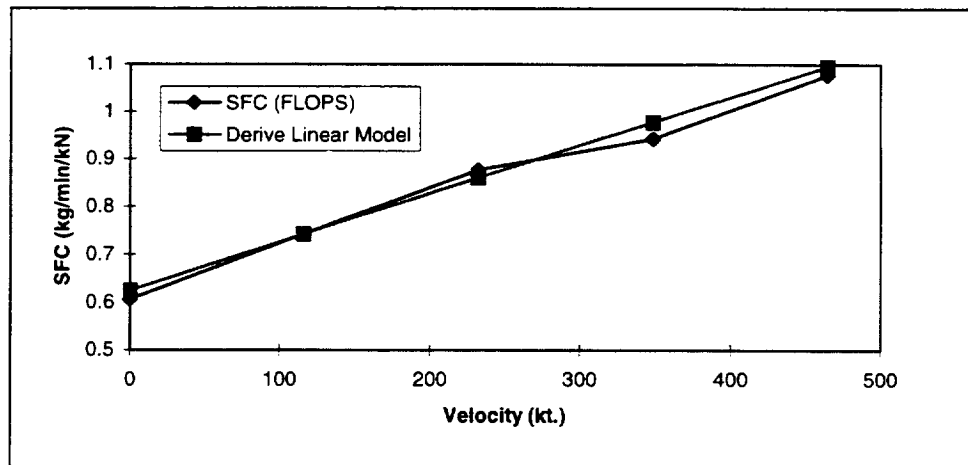
An example is shown in Table A-4-6.

*Table A-4-6. Sample Computation of Velocity Correction Coefficient Cf2*

Mach	Velocity	Altitude	PC	Thrust (lb)	RD FF	SFC lb/hr/lb	SFC kg/min/kN	SFC Model	Slope	Intercept
0.8	464.5516	30000	1	11473.4	7268.4	0.6335	1.076722	1.095173	0.001158	0.538956
0.6	348.4137	30000	1	10622.4	5889	0.5544	0.942281	0.977642	0.000562	0.746482
0.4	232.2758	10000	1	25046.9	12924.1	0.516	0.877014	0.86011	0.001157	0.608369
0.2	116.1379	5000	1	33887.3	14807.7	0.43697	0.742692	0.742579	0.001174	0.606381
0	0	5000	1	40579	14477.3	0.35677	0.606381	0.625047		

The computed coefficients for this model are  $Cf1 = .625$  and  $Cf2 = \frac{.625}{.0010} = .617.63$ .

*Figure A-4-14. Linear Model Derived from FLOPS.*



The descent fuel flow coefficients can be determined in the same fashion, using the descent power settings.

## PROPULSION: THRUST MODEL

Thrust for jet engines are given as a quadratic function of altitudes. The equation has three coefficients for altitude correction:

$$T_{\max climb} = C_{t1}(1 - h / C_{t2} + C_{t3}h^2)$$

Name: **Ct1**

Description: Thrust coefficient (N)

Location: DERIVED / Output file

Details: Propulsion output section.

Name: **Ct2**

Description: Thrust coefficient (First-order altitude correction) (ft.)

Location: DERIVED / Output file

Details: Propulsion output section.

Name: **Ct3**

Description: Thrust coefficient (Second order altitude correction) (1/ft<sup>2</sup>)

Location: DERIVED / Output file

Details: Propulsion output section.

### Steps for Creating a Thrust Model From FLOPS Output.

The thrust model is a quadratic function of the altitude. The Mission Generator aircraft file assumes that the thrust model is for the whole aircraft, so thrust values are the total components of all engine output. With this in mind, as the FLOPS output is translated, be sure to multiply the thrust from FLOPS by the number of engines on the vehicle.

1. Run the FLOPS model with several propulsion output points with variations in altitude at a selected Mach number proper for the chosen aircraft.
2. As in the case when constructing the fuel flow model, collect the data as an array of thrust and altitude points for the selected max power settings, converting from lbs to Newtons.
3. Ct1 will be the reference thrust for sea level. This value should be shown in the FLOPS propulsion output. This value should be converted to Newtons, and should be multiplied by the number of engines on the aircraft.

$$T_{\max \text{ climb}}(0) = C_{t1}$$

Example from the propulsion section of output file:

```
SEA LEVEL STATIC MAXIMUM THRUST      43700.0  LB
      (NEW BASELINE ENGINE THRUST - OVERRIDES THRSO)

# REVISED PROPULSION SYSTEM DATA
      BASELINE ENGINE THRUST    THRSO      43700.0  LBF
```

1. In order to obtain Ct2 and Ct3, we will have to use two data points and solve two equations.  
First, we can rewrite the thrust equation, multiplying out the Ct1 and renaming the coefficients

$$T(h1) = (C1 - C2h1 + C3h1^2)$$

$$T(h2) = (C1 - C2h2 + C3h2^2)$$

We can rewrite C2 as a function of C3 from equation 1, and then solve equation 2 for C3.

$$C2 = \left[ \frac{C1 - T(h1) + C3h1^2}{h1} \right]$$

$$C3 = \frac{\left[ C1 \left\{ 1 - \frac{h2}{h1} \right\} + T(h1) \frac{h2}{h1} - T(h2) \right]}{(h1^2 h2 - h1^2)}$$



Once C1,C2, and C3 are obtained, we can then compute Ct1, Ct2 and Ct3 from the following:

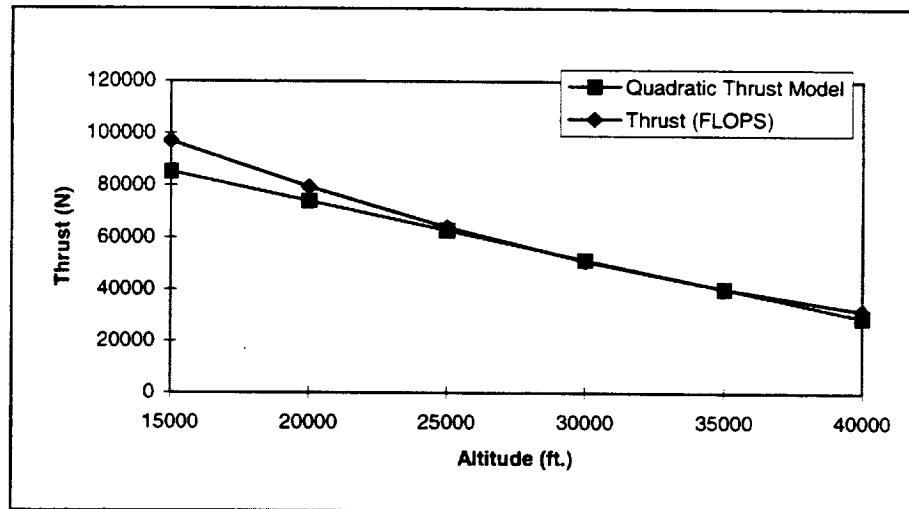
$$\begin{aligned} C1 &= Ct1 \\ C2 &= Ct1/Ct2 \\ C3 &= Ct1*Ct3 \end{aligned}$$

Table A-4-7 is an example of a portion of the FLOPS propulsion output file and the actual computed coefficients for this model.

*Table A-4-7. Sample Computation of Velocity Correction Coefficient Cf2*

Computed Coefficients		Altitude	Model	Thrust (lb)	T (N)	RAM FF	SFC	NOx Em
C1	118964.9	40000	29083.75	7122.1	31679.1	4399.4	0.61772	5.794
C2	2.247026	35000	40318.9	9064.5	40318.9	5610.1	0.61891	6.62
C3	-5.3E-11	30000	51554.04	11473.4	51033.68	7268.4	0.6335	8.777
		25000	62789.19	14373.2	63931.99	9319.1	0.64837	11.558
Ct1	118964.9	20000	74024.33	17847	79383.46	11840.1	0.66342	15.134
Ct2	52943.25	15000	85259.47	21807.1	96997.98	14797	0.67854	19.491
Ct3	-4.5E-16	10000	96794.61	23531.4	104667.7	16386.4	0.69636	21.299
		5000	107729.7	25206.2	112117.2	18078.8	0.71724	23.253
		0	118964.9	26745.7	118964.9	19862.9	0.74266	25.357

Figure A-4-15. Thrust Curve Model Constructed from FLOPS Data.



## Obtaining Fuel-Burn Rates from FLOPS Output (Trajectory)

The fuel burn rates for the aircraft input file can be derived from the FLOPS output in several ways. The easiest method is to examine the trajectory output section, and for each phase of the FLOPS trajectory, simply compute a fuel burn rate per hour. This will not be as accurate as using propulsion information, however, a rough estimate can be obtained quickly. Be careful of phases that cross too much altitude, as the result may be averaged over too much space or time.

For example, a portion of the mission output from FLOPS is shown in Figure A-4-16. For the example climb segment, we can compute the fuel burn rate (in lb./hr) as:

$$(9278. \text{ lb. } / 36.3 \text{ min}) * (60 \text{ min/hr}) = 15,335 \text{ lb./hr for climb.}$$

This is a value for the whole aircraft, and is a simple approximation based on the operating conditions reported in the trajectory output.

Figure A-4-16. Mission Summary

Initial Nox Emiss (lb)		Fuel (lb)		Time (min)		Distance (NMI)		Mach Number		Altitude (ft)	
Segmt	Weight	Segmt	Total	Segmt	Total	Segmt	Total	Start	End	Start	End
Taxi-out	256820.	475	475.	10.0	10.0						
Takeoff	256345.	237.	712.	.4	10.4				.300		0.00
Climb	256108.	8566.	9278.	25.9	36.3	186.4	186.4	.300	.800	0.	31203.
Cruise	247542.	59266.	68544.	303.4	339.7	2344.0	2530.4	.800	.739	31203.	37932.
Descent	188276.	790.	6933.	17.1	356.8	85.2	2615.6	.739	.300	37932.	0.00
(Thold		2918.	17.3)								
Reserves	187487.	6721.	76054.								
Taxi-in		475.		10.0	366.8						
Zero Fuel	180766.										
Design Range		2615.6									
Flight Time		346.4									
Block Time =	6.11 Hours										
Block Fuel =	69808.0 Pounds										
# Cruise Optimization Results											
Summary Data for Cruise Schedule 1											
Weight	Altitude	Thrust		Fuel Flow		SFC	Mach				
		Available	Required				Number				
177000.	37760.	15984.	15603.	9658.	.6190	.8000					
180000.	37443.	16230.	15842.	9807.	.6190	.8000					
183000.	37126.	16476.	16083.	9957.	.6191	.8000					
186000.	36807.	16724.	16323.	10107.	.6192	.8000					
189000.	36448.	16972.	16565.	10258.	.6193	.8000					
Summary Data for Cruise Schedule 1											
Velocity	Specific Range	Climb Rate	CL	L/D	Engine Nox	Oper Rate					
458.9	.04751	100.0	.2549	11.34	2.0	.00					
458.9	.04679	100.0	.2554	11.36	2.0	.00					
458.9	.04608	100.0	.2557	11.38	2.0	.00					
458.9	.04540	100.0	.2560	11.39	2.0	.00					
458.9	.04473	100.0	.2561	11.41	2.0	.00					



## Appendix B

# Default Data Provided With the Mission Generator

## INPUT DATA FILES—SOURCES AND AVAILABLE OPTIONS

### Aircraft Files

Aircraft files needed for the Mission Generator are produced from data contained in the \*.PTF and \*.OPF BADA (Base of Aircraft Data) files. Table B-1 shows the available aircraft.

*Table B-1. Available Aircraft*

Symbol	Manufacturer	Model	File name
ATR	ATR	ATR 42	AT42
AT7	ATR	ATR 72	AT72
707	BOEING	707-300/400 B707	
727	BOEING	727, ALL SERIES	B727
737	BOEING	737-100/200	B737
734	BOEING	737-400	B73F
733	BOEING	737-300	B73S
735	BOEING	737-500	B73V
747	BOEING	747-100/200/300	B747
744	BOEING	747-400	B74F
757	BOEING	757, ALL SERIES	B757
767	BOEING	767, ALL SERIES	B767
BA1	BAE	111, ALL SERIES	BA11
J31	BAE	JETSTREAM 31	BA31
J41	BAE	JETSTREAM 41	BA41
146	BAE	146-100/200/300/RJ	BA46
ATP	BAE	ADVANCED TURBOPROP	BATP
B20	BEECH	SUPER KING AIR 200/HURON	BE20
B90	BEECH	KING AIR 90	BE90
BE9	BEECH	AIRLINER C99	BE99
130	LOCKHEED	HERCULES	C130
CN4	CESSNA	GOLDEN EAGLE 421	C421
CN5	CESSNA	CITATION II-S2	C550

*TableB-1. Available Aircraft (Continued)*

Symbol	Manufacturer	Model	File name
CAN	CESSNA	CITATION V	C560
CRJ	CANADAIR	CHALLENGER 600/601	CL60
D28	DORNIER	DO 228-100/200	D228
D38	DORNIER	328	D328
DA0	DASSAULT-BREGUET	MERCURE 100	DA01
DA1	DASSAULT-BREGUET	FALCON 10	DA10
DA2	DASSAULT-BREGUET	FALCON 20 FJF/C/D/E/F	DA20
DA5	DASSAULT-BREGUET	FALCON 50	DA50
DA9	DASSAULT-BREGUET	FALCON 900	DA90
D10	MCDONNELL-DOUGLAS	DC-10,ALL SERIES	DC10
DC8	MCDONNELL-DOUGLAS	SUPER DC-8, ALL SERIES	DC8S
DC9	MCDONNELL-DOUGLAS	DC-9 /20/30/40/50	DC9
DH8	DE HAVILLAND	DASH 8 DHC-8	DH8
EMB	EMBRAER	BRASILIA EMB-120/HH/RT	E120
E3A	BOEING	E-3A SENTRY	B707
EB3	AIRBUS	A300	EA30
310	AIRBUS	A310	EA31
325	AIRBUS	A320	EA32
330	AIRBUS	A330	EA33
340	AIRBUS	A340	EA34
FGT	GENERIC	MILITARY FIGHTER	FGTR
100	FOKKER	100	FK10
F27	FOKKER	FRIENDSHIP F27	FK27
F28	FOKKER	FELLOWSHIP F28	FK28
F50	FOKKER	50	FK50
F70	FOKKER	70	FK70
HS2	BAE	BAE 125-400/600/700/800	HS25
L10	LOCKHEED	TRISTAR L1011, ALL SERIES	L101
LR3	BOMBARDIER	LEARJET 35	LR35
M11	MCDONNELL-DOUGLAS	MD-11	MD11
M80	MCDONNELL-DOUGLAS	MD-80/81/82/83/83/88	MD80
MU2	MITSUBISHI	MARQUISE/SOLITAIRE	MU2
N16	AEROSPATIALE	TRANSALL C160	ND16
PAG	PIPER	CHEYENNE III/IV,400SL	PA42
SF3	SAAB-SCANIA	SF 340	SF34
SH6	SHORTS	SH3-36	SH36

TableB-1. Available Aircraft (Continued)

Symbol	Manufacturer	Model	File name
SWM	FAIRCHILD	MERLIN IVC, METRO III	SW3
TB2	AEROSPATIALE	TRINIDAD TB-20	TB20
TU3	TUPOLEV	TU-134,134A/B	TU34
TU5	TUPOLEV	TU-154,154A/B/B2/C/M	TU54

The “file name” column in Table B-1 consists of the names of the aircraft files in the Mission Generator software, and would also be the file names produced by the bada2ac utility program. These file names do not include the “.ac” extension, which is part of the file names in the Mission Generator software in the directory UTIL/BADA2AC.

## Flight Phase Data Files

Data files are provided for the departure, taxi out, takeoff, climb, cruise (delay part only), approach, and taxi in phases of each flight. Many of these files are populated with real data for the appropriate airline, aircraft, and/or airports. Other files are provided with all zero data, and could be populated with real data as necessary models become available.

Table B-2 shows the file name, the source of the data contained in that file, and the key(s) for the data. The key(s) is the first field in the file, and gives access to the desired value(s). Files listed are in the LOOKUP directory.

## Load Factors File

A data file which contains load factors based on each airline/aircraft combination is provided. This file is in the LOOKUP directory, and is called `load_factors.dat`.

*Table B-2. Flight Phase Data Files*

File name	Source of data	Key(s) for data
departure_operation.dat	N/A	origin airport
departure_delay.dat	ASQP 1995 data, as supplied by ASAC	origin airport
taxi_out_operation.dat	ASQP 1995 data, as supplied by ASAC	origin airport, airline
taxi_out_delay.dat	CODAS	origin airport, airline
takeoff_operation.dat	turboprop: 0.75 minutes jet: 1 minute	aircraft
takeoff_delay.dat	none	origin airport
climb_operation.dat	calculated as: turboprop: 10000' divided by the climb rate at 4000' for each aircraft, as given in BADA Performance Table File (PTF) jet: 23000' divided by the climb rate at 10000' for each aircraft, given in BADA Performance Table File	aircraft
climb_delay.dat	none	origin airport
cruise_delay.dat	none	destination airport
approach_operation.dat	calculated as 10000' divided by the descent rate at 6000' for each aircraft, as given in BADA Performance Table File	aircraft
approach_delay.dat	none	destination airport
taxi_in_operation.dat	ASQP 1995 data, as supplied by ASAC	destination airport
taxi_in_delay.dat	CODAS	destination airport

**Note:** (1) ASQP is "Airline Service Quality Performance"—Departure and Arrival Delays data was used above.

(2) ASAC is NASA's "Aviation System Analysis Capability."

(3) CODAS is the FAA's "Consolidated Operations and Delay System."



The source of this data is as follows:

Parameter	Source
-----------	--------

Mail/cargo load factor	
---------------------------	--

From ASQP data, according to the following formulas:

$$\text{seats} \times \frac{200 \text{ lbs.}}{\text{seat}} \times \frac{1 \text{ ton}}{2000 \text{ lbs.}} = \text{available\_passenger\_tons}$$

$$\frac{\text{available\_ton\_miles}}{\text{aircraft\_miles}} = \text{available\_tons}$$

$$\frac{\text{cargo\_mail\_ton\_miles}}{\text{aircraft\_miles}} = \text{cargo\_mail\_tons}$$

$$\text{available\_tons} - \text{available\_passenger\_tons} = \text{available\_cargo\_tons}$$

$$\frac{\text{cargo\_mail\_tons}}{\text{available\_cargo\_tons}} = \text{mail\_cargo\_load\_factor}$$

Seats	
-------	--

From ASQP data, according to the following formula:

$$\frac{\text{aircraft\_seat\_miles}}{\text{aircraft\_miles}} = \text{seats}$$

The passenger load factor was taken from OAG airport departures data.

## RELEASE NOTES

### Bad Wind Data Days

A bad wind data day is a day on which at least one “pocket” of wind (i.e., the vector for a point in airspace) has been found to have bad data. Bad wind data days currently are

- ◆ January 20 (Julian date 20),
- ◆ July 11 (Julian date 192), and
- ◆ August 10 (Julian date 222).

---

If a segment is to be flown on one of these days, then the wind data for the day preceding the bad wind data day is used.

## The Special Use Area (SUA) Map File

The detection manager loads the SUA map file when it is first called. The file is in a non-standard text format, and contains several conventions which must be followed in order to properly represent the data. An example of the file, with accompanying explanations, is provided below<sup>1</sup>:

```
//////////////////////////////////////////
//                                demomap.in
// This file has been created by the SUA map class.
// It can be used as an input file for SUA algorithms.
// The map model file format is as follows:
// It has a header section, followed by three header
// codes
// tells the algorithm that the header has been read
// in correctly. It then provides the number of points
// which are in the file. Following a listing of
// points,
// the number of spaces in the map is loaded. Each
// space
// is then represented by its characteristics, its
// number of edges, and its edges, which are repre-
// sented
// as a list of pairs of references to the point id
// numbers in the list of points described above.
//
// The file is formatted as follows:
// <number of points>
//   <ptid0> <x0> <y0> <z0>
//   .      .      .      .
//   .      .      .      .
//   <ptidn> <xn> <yn> <zn>
//
// <number of spaces>
//   <space name>
//   <space type>
//   (1-proh., 2-res., 3-warn., 4-MOA)
//   <Min. Altitude> <Max. Altitude: -1 indicates
//   unltd.>
```

---

<sup>1</sup> We use this changed font to denote actual file contents and formats throughout this document.

```
// <Start Time> <Finish Time>
// (start = stop = 0 indicates intermittent).
// <days on/off>
// <number of edges>
// <E0 head id> <E0 tail id>
// . .
// . .
// <En head id> <En tail id>
9876 6789 5555
```

The above text comprises the map's header. It provides a description of the information required to represent the map's list of master points and each of its individual SUAs. The three integers which appear at the end of the header separate the header from the body of the file.

```
16
0 1.0 1.0 0.0
1 1.0 3.0 0.0
2 -1.0 3.0 0.0
3 -1.0 1.0 0.0
4 -3.0 1.0 0.0
5 -3.0 -1.0 0.0
6 -1.0 -1.0 0.0
7 -1.0 -3.0 0.0
8 1.0 -3.0 0.0
9 1.0 -1.0 0.0
10 3.0 -1.0 0.0
11 3.0 1.0 0.0
12 0.0 0.0 0.0
13 1.0 0.0 0.0
14 2.0 0.0 0.0
15 2.0 1.0 0.0
```

This portion of the file provides the master list of points for the map. The number of points appears at the top, followed by line items which provide an ID number and the coordinates of each point. The order of the data is as follows: the ID number, followed by its longitude in degrees (minutes in decimal), its latitude in degrees, and its altitude in feet. Since the altitude of the SUAs are stored separately from their 2D vertex locations, the altitudes provided here can be arbitrary; the program ignores them.

---

The number above corresponds to the number of spaces in the map.

```
R-1404
2
10000 20000
500 2200
0 1 1 1 1 1 0
```

The data above comprise an entry of a restricted space's characteristics. The first line of the entry is its name. It must contain no white space, but can be up to 125 characters in length. The "2" below the name indicates the space's type: 1-prohibited, 2-restricted, 3-warning, and 4-MOA. Generally the spaces will either be warning or restricted areas. The next line provides the altitudes for the space. These are not used by the detection manager, but are incorporated into the current class in order to provide the capability for further enhancements to the Flight Segment Cost Model. The altitudes provided are in feet; in practice, any units can be used, provided that they correspond to the units used for the altitudes in the route data structure. The next line provides the active start and end times of the space, in GMT. The seven integers on the line below indicate the active days: this space is active Monday through Friday.

```
12
1 0
2 1
3 2
4 3
5 4
6 5
7 6
8 7
9 8
10 9
11 10
0 11
```

The data above map out the edges of the space. R-1404 contains 12 edges, each of which are represented by reference to their endpoint IDs, as given in the master point list. The head of the edge is listed first, followed by its tail. The tangent vector of the first edge, for example, starts at point 0 and ends at point 1. The list *should* be ordered in a counterclockwise fashion, as shown here. This ensures that the polygon remains topologically consistent, which will be needed should more sophisticated and robust geometry routines be required for future enhancements of the model. The list *must* be arranged such that each point is the head of exactly one segment and the tail of exactly one segment, for two reasons: 1) each space must be closed in order to

guarantee that detection checking will be valid, and 2) some of the algorithms treat the space polygon as a point list, rather than as a segment list. In other words, the lists of the edge heads and edge tails are each sufficient to describe the polygon, provided the heads and tails are consistently specified.

```
W-2508
3
0 -1
0 0
1 1 1 1 1 1 1
4
13 12
14 13
15 14
12 15
```

The data shown above represents a complete entry for a warning space. The '0-1' data in the altitude line indicates an unlimited altitude; the '0 0' data in the start and stop times indicate an intermittent warning area. Intermittent areas must be represented by zeros in the time slots, and by 1s (active) for each day of the week.

## Map File Recommendations

Currently, the map file contains 44 SUAs, each with hand-calculated coordinates. Many of these spaces are actually clumps of several smaller spaces, each with slightly different active times, controlling groups, or purposes. The map file has been provided to demonstrate the capability of the violation detection algorithm, and should provide information with an appropriate level of fidelity for the model. However, should a complete or more accurate version of the map be desired (for example, should official FAA data become available), the map file will need to be reconstructed. This is a non-trivial exercise. The following approach should help this effort:

- ◆ Use a counterclockwise path around the space to determine the ordering of the segments. This not only provides a consistent orientation for the polygon (which is good engineering practice), but also makes the segment list easier to read.

- 
- ◆ Use a 2D CAD package or other visualization tool to view the spaces as you add new spaces into the file. This will catch typographical errors and misplaced edge endpoints.
  - ◆ If a space is not a part of a continuous clump, separate its points with white spaces in the master list of points. Additionally, list the points in counter-clockwise order whenever practicable. These conventions will make it easier to check for errors.

## Appendix C

# The ASAC Flight Segment Cost Model Cost Translator User's Guide

## INTRODUCTION

The ASAC Flight Segment Cost Model (FSCM) Cost Translator accepts resource (time and fuel) data generated by the ASAC FSCM Mission Generator or analyst and produces various costs associated with flying a network of flights for a particular airline. The Cost Translator provides a flexible way of modeling airline operating costs based on cost drivers (time, fuel, or departures) and an understanding of how these drivers influence airline costs.

The Mission Generator generates Block Time and Fuel (T, F) for various segments of flights by different carriers and aircraft types as specified in the Network Description File (.NDF). The output of the Mission Generator carries Time and Fuel for each phase of every segment in the network. This output or the Network Resource File (.NRF) is fed into the Cost Translator as input. The Scenario File (.SF) contains input parameters and flags for the Cost Translator and the Cost Factors File (.CF) consists of cost driver definitions for various airlines and/or aircraft.

The output of the Cost Translator is the Network Cost File (.NCF), which assigns various Cost Factors and categories for each segment in the Network Resource File. Together the Flight Segment Cost Model—Mission Generator and Cost Translator provide the analyst with optimized routes, block times and fuels, as well as the costs associated in flying routes in various categories. Figure C-1 shows model inputs and outputs.

## RUNNING THE MODEL

Executing the model is simple. Using Netscape™ or Internet Explorer™, go to the ASAC home page at <http://asac.lmi.org/>. From there, follow the pages until you get to the “model wizard.” From the model wizard, select “ASAC Flight Segment Cost Model—Cost Translator,” as shown in Figure C-2.

Figure C-1. Inputs/Outputs of Model

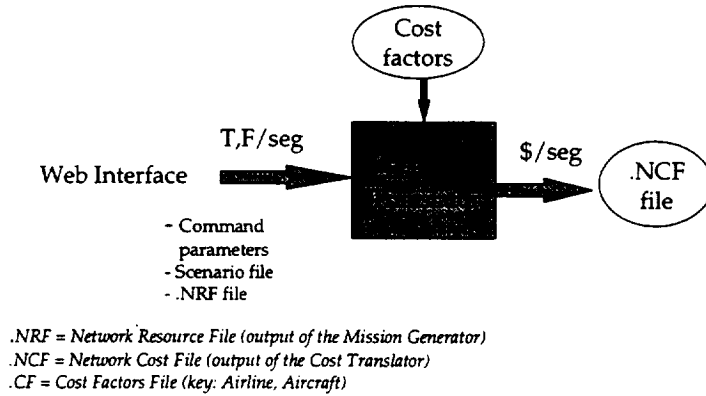


Figure C-2. Entering the FSCM Cost Translator via the World Wide Web

## ASAC Model Wizard

This will allow you to select a model or models to run, choose all appropriate options for the model or models chosen, then run the models and notify you of the results according to your selection.

[Click here for help or complete documentation](#)

**Select a model to run  
then press CONTINUE:**

ASAC Airport Capacity Model	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-bottom: 5px;">↑</div> <div style="margin-bottom: 5px;">↓</div> </div>
ASAC Airport Delay Model	
ASAC Flight Segment Cost Model - Mission Generator	
ASAC Flight Segment Cost Model - Cost Translator	
ASAC Air Carrier Network Cost Model	



A series of World Wide Web pages will take you through the process of loading the required model inputs.

## PREPARING THE INPUTS

The inputs to the Cost Translator are a Network Resource File (.NRF file), a Scenario File (.SF file), and a Cost Factors File (.CF). The Scenario File and the Network Resource File are direct user inputs. The Cost Factors File location is specified within the Scenario File values.

### Network Resource File

The Network Resource File (.NRF) is generated by the Mission Generator (or user) and serves as a primary input of the Cost Translator. Appendix A contains .NRF specifications and a sample Network Resource File. The Cost Translator expects the .NRF to have three sections (mission generator program execution summary, Scenario File used, and detailed results), but to use the detailed results section primarily. The first line in the detailed results section contains the column names of all outputs of the Mission Generator<sup>1</sup>, as shown below<sup>2</sup>:

```
SEG_NAME  ORIG DEST EQUIP  AIRLN  START  END  DEP
ARR  BLK  FL/DAY  LOAD_FACTOR  GROUND_STATUTE_MILES
PH1_OP_TM PH1_OP_FL PH1_DL_TM PH1_DL_FL PH1_TOT_TM
PH1_TOT_FL  PH2_OP_TM PH2_OP_FL PH2_DL_TM PH2_DL_FL
PH2_TOT_TM  PH2_TOT_FL  PH3_OP_TM PH3_OP_FL
PH3_DL_TM PH3_DL_FL PH3_TOT_TM  PH3_TOT_FL
PH4_OP_TM PH4_OP_FL PH4_DL_TM PH4_DL_FL PH4_TOT_TM
PH4_TOT_FL  PH5_OP_TM PH5_OP_FL PH5_DL_TM PH5_DL_FL
PH5_TOT_TM  PH5_TOT_FL  PH6_OP_TM PH6_OP_FL
PH6_DL_TM PH6_DL_FL PH6_TOT_TM  PH6_TOT_FL
PH7_OP_TM PH7_OP_FL PH7_DL_TM PH7_DL_FL PH7_TOT_TM
PH7_TOT_FL  TOT_OP_TM TOT_OP_FL TOT_DL_TM TOT_DL_FL
TOT_MNTM  TOT_VRTM  TOT_MNFL  TOT_VRFL
```

The SEG\_NAME uniquely identifies the segment which is to be costed. The AIRLN (airline) and EQUIP (equipment) headings make up the unique identifier for searching the Cost Factors File during Cost Translator execution. TOT\_MNTM (block time) and TOT\_MNFL (fuel) are the primary cost drivers used by the Cost Translator.

---

<sup>1</sup> See Appendix A, "ASAC FSCM Mission Generator User's Guide" for a description of these data elements.

<sup>2</sup> We use this changed font to denote actual file contents and formats throughout this document.

---

The number of departures for each segment, known as D, is a third cost driver available to the model. It is computed as:

$$FL/DAY \times (END-START + 1)$$

where

FL/DAY = flights per day,

END = the end date of the schedule for that flight, and

START = the start date of the schedule for that flight.

## Scenario File

The Scenario File contains names of data files and flags which control the execution parameters of the Cost Translator. The Scenario File consists of a set of tags which describe various sections of the input file. The text inside the tags [] is case-insensitive.

The following shows a sample Scenario File:

```
[Path]
$DATA = /opt/ASAC/data/CT
$WORKING = /users/special_user
[Cost_Factors]
cost_data = $DATA/default.cf

[Execution_Mode]
standalone_mode = FALSE

[Network_mode]
network_mode = TRUE
```

The [Path] tag denotes the directories that the Cost Translator will use to look for data files or to output the Network Cost File. This path is generated automatically by the web interface.

The [Cost\_Factors] cost\_data tag denotes the location of the Cost Factors File. A default file is provided, although the user may upload his or her own file, or modify the default file via the world wide web (see Figure C-2).

The [Execution\_Mode] standalone\_mode tag tells the Cost Translator whether it is being run in conjunction with the Network Cost Model. The default value, FALSE, indicates that the Network Cost Model will run afterwards. A TRUE value indi-

cates that the Network Cost Model will not be run. This has implication for Cost Translator outputs.

The [Network\_mode] network\_mode tag denotes whether segment resource data is in the .NRF (Network Resource File) format. The default value is TRUE; i.e., it is assumed that the Mission Generator has run or that the user has generated a Network Resource File that is consistent with the Mission Generator output format. *Note: The alternate scenario, where network\_mode is FALSE, is reserved for future use.*

## Cost Factors File

The Cost Factors File provides an extremely flexible mechanism for defining cost drivers, factors or sub-categories, and categories, and relationships between them. Some important term definitions follow:

- ◆ *Drivers* of the Cost Translator are block (T)ime (hours), (F)uel (pounds), and (D)epartures. Not all of these need to be used in any one Cost Factors File.
- ◆ *Factors* or *Sub-Categories* are dependent upon drivers (e.g., labor). Factors are defined as the sum of a constant and a coefficient for each driver time that driver. For example,

$$\text{LABOR} = a + b_T.T + b_F.F + b_D.D.$$

Constant values and coefficients are defined in the table for each air-line/equipment combination.

- ◆ *Categories* (e.g., Variable Operating Cost) are dependent upon factors and are the sum of a constant and one or more factors. For example:

$$\text{VOC} = \text{LABOR} + \text{FUEL\_COST} + \text{MAINTENANCE} + \text{OTHER}.$$

*Note:* All names must not contain spaces. Spaces indicate a change in field.

## What is in the Cost Factors File

A sample Cost Factors File (.CF) is shown below. (Comment lines begin with "//.")

```
NUM_DRIVERS=3
NUM_FACTORS=4
NUM_CATEGORIES=2
// ***** DRIVERS *****
```

---

```

DRIVERS= T F D
// T = BLOCK TIME
// F = BLOCK FUEL
// D = DEPARTURES
// ***** FACTORS *****
FACTORS= L M OWN OTH
// L = Labor
// M = Maintenance
// OWN = Ownership
// OTH = Other
//***** CATEGORIES *****
CATEGORIES=VOC DOC
// VOC = Variable Operating Cost less Fuel
// DOC = Direct Operating Cost
VOC=L M OWN
// VOC = Factor1 + Factor2 + ... + Factor n
VOC_CONST=1000
// The Constant value for VOC calculation goes above
DOC=L M OWN OTH
// DOC = Factor1 + Factor2 + ... + Factorm
DOC_CONST= 500
// The Constant value for VOC calculation goes above
=====
// ***** DATA DEFINITION *****
// Here all the FACTORS are defined in terms of the
DRIVERS above. For each FACTOR, // we have to define
the numerical constant first and then one coeffi-
cient for each DRIVER.
// E.g., the FC cost is calculated as:
//  $L = a + b_T.T + b_D.D + b_F.F$ 
// where a = constant
// b_j = coefficient for DRIVER j
// The following columns carry the a and b's for
each FACTOR of each Airline and Equipment
// We always need a default airline and equipment
data line, for cases where the input file
// (e.g., Network Resource File or Network Average
Statistics file) contains airline and equipment
// pairs with no distinct match with this file.
// ARLN EQUIP L_CONST L_T L_F L_D ...OTH_CONST OTH_T
OTH_F OTH_D
//
UA 747 500 1000 0 0 0 1000 0 0 900 40
0 2000 3.45 4.56 5.67 7.89
UA 757 400 1 343 2 3 3424 3 3 123

```

34	1	4334	1.23	3.45	2.45	5.72						
UA	DEF	342	4	34	344	3	234	43	3	432	34	
4	3242	4.67	7.89	3.67	5.89							
DF	DEF	4343	4	23	344	8	432	43	3	88		
98	8	168	5.6	12.4	3.67	3.88						

The first part of the Cost Factors File declares the number of drivers, factors, and categories defined in the file:

```
NUM_DRIVERS=3
NUM_FACTORS=4
NUM_CATEGORIES=2
```

The second part identifies the three drivers to be used as noted by the NUM\_DRIVERS:

```
DRIVERS= T F D
// T = BLOCK TIME
// F = BLOCK FUEL
// D = DEPARTURES
```

*Note: The Cost Translator currently recognizes these three cost drivers only and computes their values for each segment from the .NRF input data.*

The third part defines the factors in terms of the drivers. Any name for the factors will be accepted. The number of factors defined must match NUM\_FACTORS above:

```
FACTORS= LABOR MAINT OWN OTH
// LABOR = Labor
// MAINT = Maintenance
// OWN = Ownership
// OTH = Other.
```

Finally, the cost categories are identified and defined in terms of the factors above. Any name for the cost categories is allowed. Again, the number of categories must match the NUM\_CATEGORIES above:

```
CATEGORIES=VOC DOC
// VOC = Variable Operating Cost
// DOC = Direct Operating Cost
VOC=LABOR MAINT OWN
// Category = Factor1 + Factor2 +...+ Factor n (with an
implicit plus sign).
VOC_CONST=1000
// The constant value for VOC calculation goes above.
It could be zero, integer, and float.
```

UA	747	500	1000	0	0	0	1000	0	0	900	40
0 2000	3.45	4.56	5.67	7.89							

labor = \$500 + \$1000/block hour (plus \$0/# fuel plus \$0/departure)  
maintenance = \$1000/# fuel

Appendix D contains details on how default Cost Factorss were derived.

The Cost Translator reads each segment in the .NRF, echoes all the columns to the .NCF, extracts relevant values from the .NRF for that segment, reads relevant constants and coefficients from the Cost Factors File (.CF), and outputs various cost figures for that segment to the output, a Network Cost File (.NCF). Part of a sample Network Cost File is shown below with new fields added to the Network Resource File by the Cost Translator shown in bold:

C-8

```

0.02 340 0.45 7650 0 0 0.45 7650 2.45509 17512.4 0
0 2.45509 17512.4 0 0 0 0 0 0 0.1 140 0.1
140 2.94509 25530.4 0.245656 343.918 3.19075
0.00515257 25874.3 422022 3501.31 3104.92
1570.81 52.33 8229.36

ORDMIA020040 ORD MIA 757 UA 20 40 7:30 12:41 3:35
0.8 0.66 1.1e+03 0 0 0.19 2.6e+02 0.19 2.6e+02 0.27
3.8e+02 0 0 0.27 3.8e+02 0.02 3.4e+02 0 0 0.02
3.4e+02 0.45 7.6e+03 0 0 0.45 7.6e+03 2.2 1.6e+04 0
0 2.2 1.6e+04 0 0 0 0 0 0.11 1.6e+02 0 0 0.11
1.6e+02 3.1 2.4e+04 0.19 2.6e+02 3.3 0.0042 2.5e+04
2.8e+05 3597.67 2943.06 1614.04 53.77 8208.54

```

## Analyzing the Model Output

The output of the Cost Translator adds all the cost columns (factors and categories) for each segment in the .NRF file to the Network Cost File (.NCF). Look at the column headers in **BOLD** to identify the model output. The number of cost columns will vary with the number of factors and categories defined in the Cost Factors File (.CF) by the user. The cost columns start to the right of the "TOT\_VRFL" column.

The total of the cost columns can be easily calculated via a spreadsheet. The total would apply to the entire network of flight segments. If the "standalone\_mode" is false (which is the default value), then it is assumed that the Network Cost Model will run next and each cost column will reflect the cost for flying the segment one time. This is done because the Network Cost Model will run later and weight the cost columns by "D," or number of departures.

If the "standalone\_mode" is true, then the cost columns of this model will reflect the cost of flying the segment D times, where  $D = FL/DAY \times (END-START + 1)$ .





## Appendix D

# Flight Segment Cost Model Cost Translator Default Data

The primary default data for the Cost Translator is the Cost Factors File (see Appendix C). The default Cost Factors File for ASAC addresses those airline costs that comprise variable operating costs plus an allocated cost for flight attendant labor.

## DEFINITION OF VARIABLE OPERATING COST

Traditionally, operating costs in the airlines industry have been discussed in terms of direct and indirect categories. Direct costs are associated with flying the aircraft and indirect costs are associated with serving the passengers. Direct and indirect costs may also have fixed and variable components. For example, cyclic airframe maintenance is a fixed direct operating cost while fuel burned is a variable direct operating cost. Conversely, reservation systems are a fixed indirect operating cost and passenger in-flight meals are a variable indirect operating cost. Therefore, direct and indirect operating costs are not synonymous with fixed and variable costs.

The Department of Transportation Form 41 data, the Aircraft Synthesis model, the Flight Optimization model, and the National Airspace System Performance analysis Capability “Cost of Delay” module all use a taxonomy of direct and indirect operating costs. Fortunately, there is a close match among the cost categories from all four sources. This facilitates comparisons across the three models and benchmarking against the airlines’ actual expenses.

For purposes of the ASAC, it is useful to consider an airline’s core business as a network comprised of takeoff and landing slots at the airports from and to which it chooses to fly. In the short run, the size of this network is fixed. Consequently, the costs associated with having the total capacity (the aircraft, maintenance facilities, passenger and cargo handling facilities, etc.) required to operate this network are also constant in the short run. Examples of this category of airline expense, which are “fixed” operating costs, include aircraft rentals and interchange charges, aircraft and traffic service, passenger service (including cabin crew expenses), promotion and sales, maintenance of ground property and equipment, depreciation and amortization, other transportation related expenses, and general and administrative costs.

Conversely, some costs are dependent upon how intensively the network capacity is used, e.g., how frequently and efficiently planes move between the airports served by the network. Examples of this type of airline expense, which we term “variable”

---

operating costs (VOC), include salary, benefits, payroll taxes, and other personnel expenses for the flight crew; aircraft fuel and oil (including fuel and oil taxes); flight equipment maintenance (including maintenance overhead or burden); and insurance, injuries, loss, and damage.

## SOURCE OF AIRLINE OPERATING COSTS DATA

The Department of Transportation collects data from large air carriers (also known as Form 41 Carriers). It is empowered to require annual, monthly, periodical, and special reports from any air carrier; to prescribe the manner and form in which such reports shall be made; and to require from any air carrier specific answers to all questions upon which information is deemed to be necessary. This data includes

- ◆ *P-5.1 aircraft operating expenses*—filed by group I carriers by entity and equipment type, either quarterly (for carriers with annual revenues over \$20 million) or semi-annually (for annual revenues below \$20 million). The data include general cost categories of pilot and copilot, aircraft fuel and oil, other flying expenses, maintenance of flight equipment, and depreciation and rentals of flight equipment. In general, these cost elements are reported at a much higher level of aggregation compared with the P-5.2 report.
- ◆ *P-5.2 aircraft operating expenses*—filed by group II and III carriers by entity and equipment type on a quarterly basis. The data include the general cost categories of flying operations, direct maintenance of flight equipment, applied maintenance burden for flight equipment, net obsolescence and deterioration of expendable parts, depreciation of flight equipment, and expense of interchange aircraft.
- ◆ *P-6 operating expenses by objective groupings*—filed by group I carriers with annual revenues over \$20 million and by group II and III carriers at the entity level only (not by equipment type). The quarterly data include cost elements in the general categories of salaries and related fringe benefits, materials purchased, services purchased, landing fees, rentals, depreciation, amortization, transport related expense, and other expenses.
- ◆ *P-7 operating expenses by functional groupings*—filed by group III carriers at the entity level only (not by equipment type). The quarterly data include cost elements in the general categories of aircraft operating expenses (direct operating costs), passenger service, aircraft servicing, traffic servicing, reservations and sales, advertising and publicity, ground property and equipment, and general and administrative.

- ◆ *T-2 traffic, capacity, aircraft operations, and miscellaneous statistics*—filed by all carriers by entity and equipment type. The quarterly data include such elements as revenue passenger miles, available seat miles, aircraft miles, block hours, flight hours, aircraft days, and gallons of fuel.

Because cost data are available on an equipment basis and at a fine enough level of detail only from the P-5.2 report, we were able to obtain detailed costs on those carriers only. Table D-1 shows the VOC cost categories and the Form 41 cost elements that compose them.

*Table D-1. Variable Operating Cost Categories and Elements*

Category	Form 41 components	Description
Personnel	5123.0, 5124.0, 5128.1, 5136.0, 5157.0, 5168.0	Pilots and copilots, other flight personnel, trainees and instructors, personnel expenses, employee benefits and pensions, payroll taxes
Fuel and oil	5145.1, 5145.2, 5169.0	Aircraft fuel, aircraft oil, taxes other than payroll
Maintenance	5225.1, 5225.2, 5243.1, 5243.2, 5243.7, 5246.1, 5246.2, 5272.1, 5272.3, 5272.6, 5272.8, 5279.6	Airframe labor, engine labor, airframe outside repair, engine outside repair, interchange maintenance, airframe materials, engine materials, airframe airworthiness provisions, airframe deferred overhauls, engine airworthiness provisions, engine deferred overhauls, applied maintenance burden-flight equipment
Other	5155.1, 5158.0, 5153.0, 5171.0	General insurance purchased, injuries loss and damage, other supplies, other expenses

## DEFAULT COST FACTORS

Default cost factors were obtained by dividing the total costs in the VOC categories, except fuel, by block hours flown for eight major airlines (American Airlines, Continental Airlines, Delta Airlines, Northwest Airlines, Southwest Airlines, Trans World Airlines, United Airlines, and USAir) and the equipment they flew in 1994 (the last year with cost data readily accessible). Additionally, all other airlines were combined into a pseudo-airline given the arbitrary code of "LC." These data could be useful if an analyst was interested particularly in the cost structure of smaller airlines.

Data on all carriers were combined to generate a set of default data for individual equipment types. This data can be used if an analyst chooses to look at systemic costs rather than the costs of a particular airline.

Data on all equipment flown by a given airline were combined to generate default costs for that airline.

The Boeing 737-300 (very common on domestic flights) was used to generate a default aircraft for an unspecified airline (default) so that the model would not fail given the lack of a recognized airline/aircraft combination.

Table D-2 summarizes the default data provided for use with the cost translator. Note, all airlines are assumed to have a fuel cost of \$0.12 per pound of fuel.

## OTHER DEFAULT AND SAMPLE FILES

Also provided with the model are a default Scenario File which uses the default Cost Factors File and a sample Network Resource File for flights from O'Hare International Airport in Chicago to Miami International Airport and back. The default Scenario File allows the user to minimize model input to the flight segment resource information (the Network Resource File); the sample Network Resource File allows the user to become familiar with the model.

*Table D-2. Default Cost Factors*

Airline	Equipment	Labor \$/hour	Fuel \$/#	Maintenance \$/Hour	Other \$/hour
AA	100	758.26	0.12	203.36	20.39
US	100	768.11	0.12	282.85	14.49
DF	100	761.81	0.12	231.96	18.27
LC	146	400.59	0.12	589.29	61.31
DF	146	400.59	0.12	589.29	61.31
LC	727	1058.01	0.12	1390.06	109.37
DF	727	1058.01	0.12	1390.06	109.37
CO	733	538.25	0.12	342.48	12.64
DL	733	817.05	0.12	445.61	8.95
UA	733	728.31	0.12	475.22	9.18
US	733	827.13	0.12	524.80	19.66
WN	733	555.79	0.12	367.01	7.94
DF	733	678.22	0.12	435.78	12.12

Table D-2. Default Cost Factors (Continued)

Airline	Equipment	Labor \$/hour	Fuel \$/#	Maintenance \$/Hour	Other \$/hour
LC	734	619.55	0.12	295.22	37.11
US	734	836.66	0.12	184.26	23.55
DF	734	752.83	0.12	227.11	28.79
CO	735	512.38	0.12	78.38	1.14
UA	735	690.12	0.12	451.99	19.44
WN	735	534.34	0.12	283.74	8.53
DF	735	637.85	0.12	394.51	15.74
DF	737	647.78	0.12	438.86	8.92
CO	743	1479.22	0.12	2722.28	17.01
LC	743	1896.83	0.12	2582.98	217.21
UA	743	2265.55	0.12	2211.09	30.36
DF	743	2009.98	0.12	2427.15	98.94
NW	744	2230.65	0.12	685.17	61.57
UA	744	2374.83	0.12	1301.78	48.55
DF	744	2335.39	0.12	1133.12	52.11
CO	747	1464.97	0.12	1687.06	7.52
LC	747	1035.32	0.12	1660.31	140.04
NW	747	1986.27	0.12	1519.95	20.28
TW	747	1720.11	0.12	1621.36	22.77
UA	747	2179.14	0.12	1323.00	9.12
DF	747	1692.97	0.12	1540.03	53.71
AA	757	980.06	0.12	355.10	10.98
CO	757	732.54	0.12	231.27	15.08
DL	757	988.34	0.12	512.31	15.19
LC	757	719.39	0.12	523.79	81.01
NW	757	888.35	0.12	331.33	16.03
UA	757	1097.33	0.12	492.30	16.40
US	757	1085.92	0.12	434.04	33.48
DF	757	979.23	0.12	449.06	23.27
AA	763	1136.54	0.12	378.79	5.14
DL	763	1307.75	0.12	491.09	19.98
TW	763	1052.41	0.12	811.54	19.52
UA	763	1214.27	0.12	246.22	0.47

Table D-2. Default Cost Factors (Continued)

Airline	Equipment	Labor \$/hour	Fuel \$/#	Maintenance \$/Hour	Other \$/hour
DF	763	1215.39	0.12	395.79	9.63
AA	767	1017.89	0.12	612.10	5.70
DL	767	1071.77	0.12	442.00	19.37
TW	767	955.07	0.12	656.00	22.55
UA	767	1116.28	0.12	968.23	42.35
US	767	1515.14	0.12	935.96	36.30
DF	767	1089.51	0.12	709.02	21.95
LC	32S	550.32	0.12	278.42	41.41
NW	32S	794.63	0.12	209.02	17.27
UA	32S	854.30	0.12	251.74	13.76
DF	32S	747.18	0.12	231.62	22.32
LC	72S	866.14	0.12	1112.38	104.49
AA	72S	1004.08	0.12	437.14	16.10
CO	72S	670.70	0.12	561.85	5.38
DL	72S	1060.51	0.12	571.93	2.07
NW	72S	913.83	0.12	496.6	5.16
TW	72S	644.76	0.12	565.72	9.45
UA	72S	969.16	0.12	764.29	4.25
US	72S	1428.63	0.12	877.85	21.68
DF	72S	920.97	0.12	652.58	22.38
LC	733	480.15	0.12	257.04	36.12
CO	737	471.86	0.12	604.82	1.60
DL	737	754.82	0.12	397.42	5.24
LC	737	468.77	0.12	444.73	28.55
UA	737	677.01	0.12	474.36	1.59
US	737	840.23	0.12	496.67	6.71
WN	737	528.26	0.12	286.45	8.41
CO	73S	471.86	0.12	604.82	1.60
DL	73S	754.82	0.12	397.42	5.24
LC	73S	468.77	0.12	444.73	28.55
UA	73S	677.01	0.12	474.36	1.59
US	73S	840.23	0.12	496.67	6.71
WN	73S	528.26	0.12	286.45	8.41

*Table D-2. Default Cost Factors (Continued)*

Airline	Equipment	Labor \$/hour	Fuel \$/#	Maintenance \$/Hour	Other \$/hour
DF	73S	647.78	0.12	438.86	8.92
NW	74F	1045.33	0.12	1690.40	28.79
DF	74F	1045.33	0.12	1690.40	28.79
UA	74L	1887.50	0.12	2190.64	24.81
DF	74L	1887.50	0.12	2190.64	24.81
LC	AB3	757.77	0.12	555.80	112.35
AA	AB3	1224.28	0.12	931.28	11.96
CO	AB3	815.43	0.12	1169.10	17.00
DF	AB3	1053.36	0.12	1020.60	15.54
LC	AT7	193.43	0.12	173.58	10.93
DF	AT7	193.43	0.12	173.58	10.93
LC	ATR	181.89	0.12	211.87	10.67
DF	ATR	181.89	0.12	211.87	10.67
LC	BE1	211.41	0.12	220.94	27.37
DF	BE1	211.41	0.12	220.94	27.37
LC	D10	1127.30	0.12	1971.03	129.21
LC	D10	1436.730	0.12	2063.21	220.49
AA	D10	1408.24	0.12	1077.82	8.03
CO	D10	1006.00	0.12	1325.61	18.72
NW	D10	1493.51	0.12	1264.47	8.17
UA	D10	1657.60	0.12	1778.48	11.26
DF	D10	1422.02	0.12	1541.82	47.26
LC	D38	237.05	0.12	154.57	33.41
DF	D38	237.05	0.12	154.57	33.41
LC	D9S	384.87	0.12	342.03	11.01
LC	D9S	575.29	0.12	899.49	10.96
CO	D9S	480.44	0.12	454.29	3.28
NW	D9S	627.30	0.12	387.80	3.27
TW	D9S	469.91	0.12	530.45	11.00
US	D9S	769.34	0.12	778.58	4.68
DF	D9S	605.65	0.12	525.15	5.41
LC	DC9	386.54	0.12	283.18	8.52
NW	DC9	585.15	0.12	509.35	0.78

*Table D-2. Default Cost Factors (Continued)*

Airline	Equipment	Labor \$/hour	Fuel \$/#	Maintenance \$/Hour	Other \$/hour
TW	DC9	386.06	0.12	457.68	6.22
DF	DC9	500.97	0.12	448.81	3.59
LC	DH7	465.20	0.12	860.10	10.85
DF	DH7	465.20	0.12	860.10	10.85
LC	DH8	214.87	0.12	239.92	25.40
DF	DH8	214.87	0.12	239.92	25.40
LC	EM2	133.96	0.12	207.32	9.46
DF	EM2	133.96	0.12	207.32	9.46
LC	EMB	111.17	0.12	170.65	0.74
DF	EMB	111.17	0.12	170.65	0.74
LC	F28	398.58	0.12	591.94	56.54
DF	F28	398.58	0.12	591.94	56.54
LC	J31	98.90	0.12	128.66	31.86
DF	J31	98.90	0.12	128.66	31.86
LC	L10	3538.96	0.12	6453.11	294.46
DL	L10	2049.19	0.12	1119.81	5.20
TW	L10	1102.37	0.12	1265.58	28.46
DF	L10	1789.82	0.12	1370.31	23.91
DL	L15	1547.22	0.12	1011.61	5.76
DF	L15	1547.22	0.12	1011.61	5.76
LC	M11	1261.83	0.12	1341.30	231.33
AA	M11	1328.62	0.12	725.70	5.29
DL	M11	1879.33	0.12	658.34	38.58
DF	M11	1468.20	0.12	898.75	85.74
AA	M80	795.94	0.12	349.33	14.46
CO	M80	605.45	0.12	439.93	10.96
DL	M80	818.16	0.12	250.20	10.84
LC	M80	682.52	0.12	266.28	10.82
NW	M80	763.99	0.12	393.32	9.52
TW	M80	591.90	0.12	348.43	13.75
US	M80	872.61	0.12	481.84	14.75
DF	M80	754.74	0.12	340.48	12.87
AA	M87	795.94	0.12	349.33	14.46



*Table D-2. Default Cost Factors (Continued)*

Airline	Equipment	Labor \$/hour	Fuel \$/#	Maintenance \$/Hour	Other \$/hour
CO	M87	605.45	0.12	439.93	10.96
DL	M87	818.16	0.12	250.20	10.84
LC	M87	682.52	0.12	266.28	10.82
NW	M87	763.99	0.12	393.32	9.52
TW	M87	591.90	0.12	348.43	13.75
US	M87	872.61	0.12	481.84	14.75
DF	M87	754.74	0.12	340.48	12.87
LC	SF3	187.19	0.12	226.50	16.59
DF	SF3	187.19	0.12	226.50	16.59
LC	SH6	283.87	0.12	263.52	137.39
DF	SH6	283.87	0.12	263.52	137.39
LC	SWM	138.55	0.12	173.46	15.58
DF	SWM	138.55	0.12	173.46	15.58
AA	DEF	901.35	0.12	410.25	13.36
CO	DEF	616.83	0.12	565.83	9.25
DL	DEF	1015.43	0.12	461.09	10.01
NW	DEF	976.53	0.12	589.46	11.60
TW	DEF	699.99	0.12	586.43	14.52
UA	DEF	1073.71	0.12	706.51	13.23
US	DEF	863.22	0.12	494.98	14.96
WN	DEF	541.30	0.12	322.31	8.21
DF	733	678.22	0.12	435.78	12.12



## Appendix E

# The ASAC Network Cost Model User's Guide

---

## INTRODUCTION

The ASAC Network Cost Model (NCM) accepts information on a network of flight segments and produces summary information on the traffic, resources, and operating costs associated with that network. Normally, the model receives departures, time, fuel, and cost data generated by the Flight Segment Cost Model (FSCM) and produces cost and revenue statistics for each airline/aircraft pair, as well as airline subtotals and totals for the entire network. Alternatively, the Network Cost Model is capable of running stand-alone by accepting user input on average departures and stage length for various equipment. These two distinct modes will be discussed separately.

### Network Cost Model as Part of the ASAC Model Chain

When used in conjunction with the ASAC FSCM, the Network Cost Model receives a network of individual flight segments with data on airline, equipment, departures, distances, fuel, time, and costs for each segment. The output of the ASAC FSCM Cost Translator, the Network Cost File (.NCF) is used directly as input for the Network Cost Model. The Network Cost Model also uses the same Load Factors File used by the ASAC FSCM Mission Generator to calculate flight loads (passengers and mail/cargo)<sup>1</sup>. The output of the Network Cost Model is a Network Summary File (.NSF) for each airline, by equipment type. Contained in the summary for each airline are

- ◆ equipment type,
- ◆ number of departures (flights),
- ◆ total block time,
- ◆ total fuel,
- ◆ available seat miles,
- ◆ total miles flown,

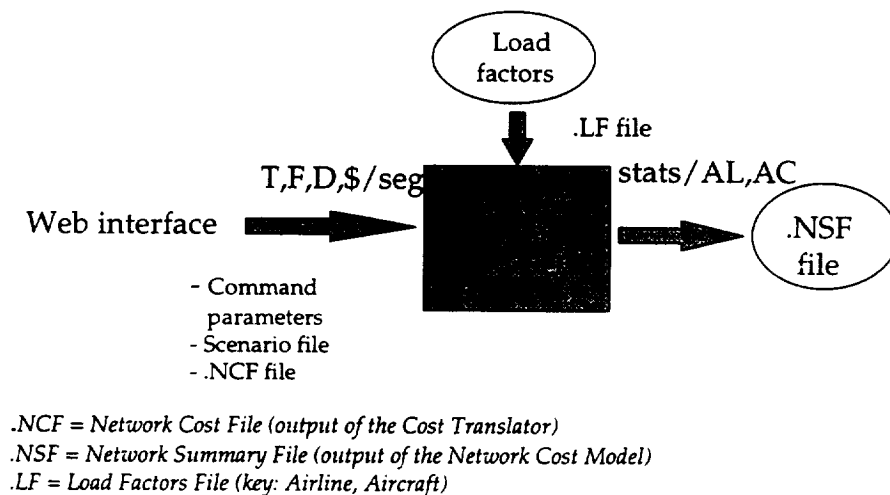
---

<sup>1</sup> See Appendix A for a discussion of the ASAC FSCM Mission Generator and Appendix C for a discussion of the ASAC FSCM Cost Translator including Network Cost File computation and format.

- ◆ revenue passenger miles, and
- ◆ costs total (by the factors and categories defined in the input Network Cost File).

Figure E-1 shows model inputs and outputs.

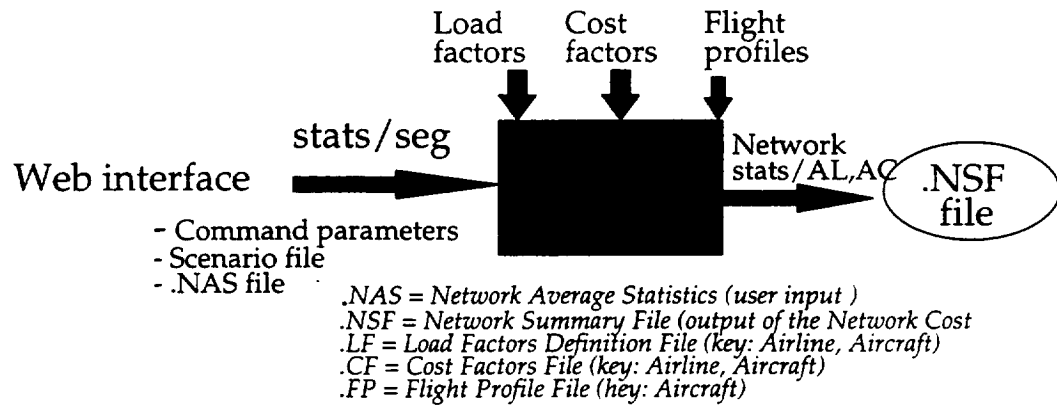
*Figure E-1. Model Inputs and Outputs, Path A*



## Network Cost Model—Stand-Alone Mode

When used in the stand-alone mode, the user provides a network average statistics file containing information on the number of departures and average stage length by each aircraft type in the network. The Network Cost Model uses a Flight Profiles File to calculate fuel and time for each “average” segment by equipment—simulating the function of the ASAC FSCM Mission Generator, but with less fidelity. It then uses the ASAC FSCM Cost Translator’s Cost Factors File to generate costs by factor and category. At this point, the Network Cost Model has generated a less detailed equivalent of the Network Cost File. Processing continues as described above, using the Load Factors File to generate the same output Network Summary File (.NSF) for each airline by equipment type. Figure E-2 shows Network Cost Model inputs and outputs when the model is run in stand-alone mode.

Figure E-2. Model Inputs and Outputs, Path B



## RUNNING THE MODEL

Executing the model is simple. Using Netscape™ or Internet Explorer™, go to the ASAC home page at <http://asac.lmi.org/>. From there, follow the pages until you get to the “model wizard.” From the model wizard, select “Network Cost Model-Cost Translator,” as shown in Figure E-3.



Figure E-3. Entering the Network Cost Model via the World Wide Web

**ASAC Model Wizard**

This will allow you to select a model or models to run, choose all appropriate options for the model or models chosen, then run the models and notify you of the results according to your selection.

Click [here](#) for help or complete documentation

Select a model to run then press **CONTINUE**:

ASAC Airport Capacity Model	 
ASAC Airport Delay Model	
ASAC Flight Segment Cost Model - Mission Generator	
ASAC Flight Segment Cost Model - Cost Translator	
ASAC Air Carrier Network Cost Model	

**CONTINUE**

A series of world wide web pages will take you through the process of loading the required model inputs.

## PREPARING THE INPUTS

Model inputs vary depending on whether the model is being run as part of the ASAC model chain or in stand-alone mode. The inputs to the Network Cost Model are the Scenario File (both modes), the Network Cost File (ASAC model chain), network average statistics (stand-alone), the Flight Profile File (stand-alone), the Load Factors File (both modes), and the Cost Factors File (stand-alone).

### Scenario File

The Scenario File contains names of data files and flags which control the execution parameters of the Network Cost Model. The Scenario File consists of a set of tags which describe various sections of the input file. Figure E-3 is a sample Scenario File for the Network Cost Model. The text inside the all the tags [] is case-insensitive<sup>2</sup>.

<sup>2</sup> We use this changed font to denote actual file contents and formats throughout this document.

```
[Path]
$DATA= ./
$WORKING= ./LOOKUP

[Execution_Mode]
stand-alone_mode = FALSE

[Load_File]
load_file_name = $DATA/newload.dat

[Cost_Factors]
cost_data= $DATA/default.cf

[Flight_Profile]
flight_profile_name=$DATA/default.fp

[Network_Average_Statistics]
network_averages_file=$DATA/default.nas
```

The [Path] tag denotes the directories that the Cost Translator will use to look for data files or to output the Network Cost File. This path is generated automatically by the web interface.

The [Execution\_Mode] stand-alone\_mode tag tells the NCM whether it is being run in conjunction with the FSCM or as a stand-alone model. The default value is FALSE, i.e., it is assumed that Cost Translator has already run and produced a Network Cost File.

The [Load\_File] load\_file\_name tag identifies the location of the Load Factors File. A default file is provided (the file used as default by the ASAC FSCM Mission Generator), although the user may upload his or her own file, or modify the default file via the world wide web. See Figures E-1 through E-3.

The [Cost\_Factors] cost\_data tag identifies the location of the Cost Factors File. A default file is provided (the file used as default by the ASAC FSCM Cost Translator), although the user may upload his or her own file, or modify the default file via the world wide web.

The [Flight\_Profile] flight\_profile\_name tag identifies the location of the Flight Profiles File. A default file is provided, although the user may upload his or her own file, or modify the default file via the world wide web.

---

The [Network\_Average\_Statistics] network\_averages\_file tag identifies the location of the Network Average Statistics File. The user may upload his or her own file, or modify an uploaded file via the world wide web.

## Network Cost File

The Network Cost File (.NCF) is generated by the ASAC FSCM Cost Translator and contains the output of the Mission Generator as well. It serves as a primary input to the Network Cost Model when it is used as part of the ASAC model chain. Appendix C contains details on this file.

## NETWORK AVERAGE STATISTICS

The Network Average Statistics File (.NAS) is created by the user and serves as a primary input to the Network Cost Model when it is used in stand-alone mode. A sample .NAS is shown below. Comments are indicated by "//."

```
//AL AC TOT_DEPARTURES ASL
//      (D)          (s.mi)
1F 747  15000      1850
1F 757  10000      1500
7V 747  10000      3000
7V 757   5000      1450
7V 767   7900      2500
AA 747  100000      1000
AA 757   80000      1500
AA 767   65000      1300
UA 737   30000      2700
UA 727   25000       900
UA 747  1000000     1200
UA 757   99000      1330
UA 767  100000      14000
YX 737   14897      15000
```

Each data line of the file contains four elements separated by one or more spaces:

- ◆ airline, normally using Official Airline Guide (OAG) codes
- ◆ aircraft equipment type, normally using OAG codes
- ◆ number of departures
- ◆ average stage length, in statute miles.



The user may not enter "DEF" for aircraft. If a new aircraft, e.g., "7x7" is entered, its corresponding data rows must be entered in the Load Factors File, Flight Profiles File, and Cost Factors File.

## Flight Profiles File

The Flight Profiles (.FP) serves as a surrogate, simplified ASAC FSCM Mission Generator for the Network Cost Model when it is used in stand-alone mode. The data below shows a sample .FP. Comments are indicated by "//."

//AC	TNC	FNC	MNC	R_TC	R_FNC
//(hr)		(lb.)	(s.mi)	(hr/mi.)	(lb./mi.)
727	0.66	100.5	500	.86	5.3
737	0.663835	100.23	575	.79	3.4
747	0.516	98.67	1000	.63	3.1
757	0.589	150.8	600	.72	4.66
767	0.516	119.23	480	.765	5.1
7x7	0.589	120.33	375	.57	4.5

Each data line of the flight contains six data elements separated by one or more spaces:

- ◆ aircraft equipment type, normally using OAG codes
- ◆ time for non-cruise phases of flight (taxi out, takeoff, climb, descent, and taxi in)
- ◆ fuel used during non-cruise phases of flight
- ◆ miles traveled during non-cruise phases of flight
- ◆ rate of block time for cruise, in hours per mile
- ◆ rate of fuel burned for cruise, in pounds per mile.

The user may not enter "DEF" for aircraft. If a new aircraft, e.g., "7x7" is entered, its corresponding data rows must be entered in the Load Factors File, Flight Profiles File, and Cost Factors File.

## LOAD FACTORS

The Load Factors File is used by the Network Cost Model in both modes (as part of the ASAC model chain and as a stand-alone model). This is the same file used by the

---

ASAC FSCM Mission Generator. See Appendix A for a detailed description. The Load Factors File is used to calculate RPM (Revenue Passenger Miles) and ASM (Available Seat Miles).

## Cost Factors

The Cost Factors File is used by the Network Cost Model in stand-alone mode. This is the same file used by the ASAC FSCM Cost Translator. See Appendix C for a detailed description.

## MODEL EXECUTION

Model execution varies significantly when the Network Cost Model is run as part of the ASAC model chain or in stand-alone mode.

### Model Execution When Run as Part of the ASAC Model Chain

In this mode, the Flight Segment Cost Model Mission Generator and Cost Translator have generated Block Time and Fuel (T, F) for various segments of flights (by carrier and aircraft type) and estimated their costs. This data is contained in the Network Cost File. Using this and the other inputs described above, the Network Cost Model performs the calculations shown in Table E-1 for each segment.

*Table E-1. Network Cost Model Calculations*

NCM columns	Name	Formula
D	Departures	$FI/Day * (End - Start + 1)$
T	Total block time	$D * TOT\_MNTM$
F	Total fuel	$D * TOT\_MNFL$
ASM	Available seat miles	$ASM = \#SEATS * D * GROUND\_STATUTE\_MILES$
TMF	Total miles flown	$TMF = \#MILES * D$
RPM	Revenue passenger miles	$RPM = ASM * LOAD$
Factor 1 \$	Cost factor 1 \$	$D * Factor\ 1\ \$$
...	...	...
Factor n \$	Cost factor n \$	$D * Factor\ n\ \$$
Category 1 \$	Cost category 1 \$	$D * Category\ 1\ \$$
...	...	...
Category m \$	Cost category m \$	$D * Category\ m\ \$$

Sources of data for these calculations are

- ◆ *output of the Mission Generator (embedded in .NCF):* flights per day, period end, period start, average block time, average fuel consumed, and miles between city pairs.
- ◆ *output of the Cost Translator (embedded in .NCF):* costs by factor and category. *Note:* The .NSF will carry the same cost factors and categories as defined in the .NCF file (Path A) or in the Cost Factors File (Path B).
- ◆ *Load Factors (.LF) File:* seats per equipment and load factor (percentage).

The summary statistics and costs of each segment are then totaled for each airline/equipment combination, with totals for each airline (all equipment types).

## Model Execution When Run as a Stand-Alone Model

When run in this mode, the Network Cost Model first estimates the resources consumed by each “average” flight (one type of aircraft with an average stage length). We have computed the fixed (non-cruise) and variable (cruise) times, fuels, and distances and documented them in the Flight Profiles File (see Appendix F). The model calculates the total time and fuel as the sum of the fixed and variable (distance-based) components. These calculations mimic the operation of the ASAC FSCM Mission Generator, but with less fidelity (they do not take into account wind, flight loads, or delays).

The calculations are performed as follows based on the Flight Profiles File data and average stage length:

$$\begin{aligned} T &= TNC + TC \\ &= TNC + R\_TC \times M\_C \\ &= TNC + R\_TC \times (ASL - MNC) \end{aligned}$$

where,

T = total time;

TNC = time for non-cruise phases, in hours;

TC = time for cruise phase, in hours;

---

R\_TC = rate of time elapsed for cruise phase, in miles per hour;

M\_C = miles for the cruise phase in miles;

ASL = average stage length, in miles; and

MNC = miles for non-cruise phases in miles.

The Network Cost Model mimics the operation of the ASAC FSCM Cost Translator, computing costs by factor and category and multiplying by number of flights (departures). It uses the Load Factors File to calculate available seat miles and revenue passenger miles. As with the use of the Network Cost Model in the ASAC model chain, output is summary statistics and costs for each airline/aircraft combination with totals for each airline.

## MODEL OUTPUT

The output of the NCM produces summaries for the network of flight segments based on unique airline and aircraft combinations. First, the NCM computes network totals for each airline/aircraft and produces cost drivers (T=time, F=fuel, D=departures), ASM, TMF, and RPM, as well as cost factors and cost categories. Second, it produces subtotals for each airline, and finally it adds a total row at the end. A partial sample Network Summary File is shown below and in Table E-2.

```
//AL AC TOT_D TOT_T TOT_F TOT_TMF TOT_ASM TOT_RPM
TOT_LABOR TOT_FUEL TOT_MAINT
TOT_OTHER TOT_VOC

AA      747 100000 51600 9867000 100000000
10000000000 7000000000 46509656 1184040
21168900 689376 69551971
AA 757 80000 51887120 347584000 120000000
24000000000 16080000000 50852490000 41710078
18425116250 569720547 69889035000
AA 767 65000 40808040 279579950 84500000
33800000000 16900000000 41538095625 33549593
24978600156 232605817 66782848438

AA TOT 245000 92746760 637030950 304500000
67800000000 39980000000 92437095281 76443711
43424885306 803015739 136741435409
```

Table E-2 shows this same output after formatting.

*Table E-2. Sample Network Cost Model Output*

Airline	Aircraft	Departures	Block time	Fuel (#)	Miles flown	ASM	RPM
AA	747	100,000	51,600	9,867,000	100,000,000	10,000,000,000	7,000,000,000
AA	757	80,000	51,887,120	347,584,000	120,000,000	24,000,000,000	16,080,000,000
AA	767	65,000	40,808,040	279,579,950	84,500,000	33,800,000,000	16,900,000,000
	Totals	245,000	92,746,760	637,030,950	304,500,000	67,800,000,000	39,980,000,000

*(Continued from above)*

Airline	Aircraft	Labor	Fuel	Maintenance	Other cost	Variable operating costs
AA	747	\$46,509,656	\$1,184,040	\$21,168,900	\$689,376	\$69,551,971
AA	757	\$50,852,490,000	\$41,710,078	\$18,425,116,250	\$569,720,547	\$69,889,035,000
AA	767	\$41,538,095,625	\$33,549,593	\$24,978,600,156	\$232,605,817	\$66,782,848,438
	Totals	\$92,437,095,281	\$76,443,711	\$43,424,885,306	\$803,015,740	\$136,741,435,409



## Appendix F

# Network Air Carrier Cost Model Default Data

---

The primary default data for the Network Cost Model are the Load Factors File, the Cost Factors File, and the Flight Profiles File. The Load Factors File is used when the model is used with either detailed flight segment information from the ASAC FSCM Cost Translator, or standalone using network average statistics. It is the same file used by the Mission Generator (see Appendices A and B). The Cost Factors File is only used by the Network Cost Model in stand-alone mode. It is the same file used by the ASAC FSCM Cost Translator (see appendices C and D).

The Flight Profiles File is used by the Network Cost Model to estimate flight resource consumption based on aircraft type and flight distances. The default data is shown in Table F-1. It was generated by flying each aircraft once to and from cruise phase to calculate the fixed time, fuel, and distance covered during non-cruise phases of flight. The cruise rates of time and fuel per mile were calculated at average altitude, mach number, and load for each aircraft.

*Table F-1. Default Flight Profiles File*

Aircraft	Non-cruise time (hours)	Non-cruise Fuel (pounds)	Distance (miles)	Cruise hours/mile	Cruise pounds/mile
100	0.59	724.6	85	0.00249	8.15
130	0.75	2236.2	80	0.00260	12.25
146	0.85	2174.1	90	0.00273	6.53
310	0.59	1324.6	110	0.00196	15.69
330	0.59	1729.7	120	0.00193	17.43
340	0.59	2259.6	130	0.00185	26.31
707	0.82	8588.5	100	0.00197	21.62
727	0.84	4589.7	105	0.00194	14.36
733	0.81	2644.9	90	0.00226	8.11
734	0.78	2634.7	90	0.00213	11.49
735	0.83	3200.1	90	0.00198	7.69
737	0.84	2946.4	90	0.00204	8.79
743	0.86	18617.5	227	0.00171	62.90
744	0.82	17631.3	227	0.00173	63.40
747	0.86	18617.5	227	0.00171	62.90
757	0.84	4931.8	105	0.00197	11.85
763	0.83	5719.9	120	0.00183	13.23
767	0.83	5719.9	120	0.00183	13.23
32S	0.59	748.0	110	0.00203	8.99
70F	0.82	8588.5	100	0.00197	21.62
72S	0.84	4589.7	105	0.00194	14.36
73S	0.81	2644.9	90	0.00226	8.11
AB3	0.59	2508.9	150	0.00181	20.50
AT7	0.79	1033.2	50	0.00358	3.55
ATP	0.83	623.2	60	0.00392	3.32
ATR	0.77	953.7	45	0.00265	2.17
B20	0.79	200.4	40	0.00314	0.73
B90	1.66	547.6	40	0.00329	1.56
BA1	0.83	2238.1	70	0.00228	6.41
BE9	1.19	300.1	70	0.00369	1.79
CNA	0.80	597.4	85	0.00249	2.55
CNA	0.84	754.5	85	0.00220	2.52
CNA	0.79	1685.7	85	0.00181	2.87
D10	0.87	10732.8	130	0.00199	22.53
D28	0.96	935.3	50	0.00371	4.77



Table F-1. Default Flight Profiles File (Continued)

Aircraft	Non-cruise time (hours)	Non-cruise Fuel (pounds)	Distance (miles)	Cruise hours/mile	Cruise pounds/mile
D38	0.77	804.0	50	0.00318	3.14
D9S	0.59	1167.2	110	0.00217	8.92
DA0	0.80	1596.5	90	0.00215	5.00
DA1	1.42	2248.8	110	0.00178	2.25
DA2	0.77	1765.7	110	0.00181	3.61
DA5	0.88	2064.8	110	0.00217	2.66
DA9	0.80	1406.7	110	0.00209	2.63
DC8	0.88	9502.4	100	0.00209	11.22
DC9	0.59	1167.2	110	0.00217	8.92
DH8	0.59	331.1	50	0.00221	1.24
EMB	0.59	389.6	50	0.00348	3.32
F27	0.59	467.5	80	0.00291	3.88
F28	0.59	1519.4	80	0.00217	6.40
F50	0.59	444.1	60	0.00414	3.47
F70	0.59	763.6	60	0.00243	6.81
HS2	0.59	779.2	100	0.00203	4.37
J31	0.78	283.0	44	0.00399	1.95
J41	0.77	474.1	39	0.00399	2.29
L10	0.59	2963.9	272	0.00221	18.69
L15	0.59	2963.9	272	0.00221	18.69
LR3	0.59	405.2	90	0.00224	2.01
M11	0.59	1690.8	130	0.00182	22.06
M80	0.59	1347.9	130	0.00237	10.09
M87	0.59	1347.9	130	0.00237	10.09
MU2	0.59	272.7	40	0.00287	1.37
N16	0.59	662.3	60	0.00310	9.38
PAG	0.59	662.3	60	0.00366	1.46
PAG	0.59	272.7	40	0.00414	1.39
SF3	0.59	1012.9	60	0.00306	3.41
SH6	0.59	317.1	70	0.00414	2.71
SWM	0.59	296.1	44	0.00392	1.36
TU3	0.59	1597.3	80	0.00191	7.27
TU5	0.59	2914.1	190	0.00185	15.33



## Appendix G

# Glossary

---

ACSYNT	Aircraft Synthesis
ASAC	Aviation System Analysis Capability
ASM	Available Seat Miles
ASQP	Airline Service Quality Performance
AST	Advanced Subsonic Technology
BADA	Base of Aircraft Data
CODAS	Consolidated Operations and Delay System
CT	Cost Translator
FAA	Federal Aviation Administration
FLOPS	Flight Optimization System
FSCM	Flight Segment Cost Model
MG	Mission Generator
MMFD	Modified Method of Feasible Directions
NASA	National Aeronautics and Space Administration
NCM	Network Cost Model
RPM	Revenue Passenger Miles
SLP	Sequential Linear Programming
SQP	Sequential Quadratic Programming
SUA	Special Use Area

REPORT DOCUMENTATION PAGE			Form Approved OPM No.0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources gathering, and maintaining the data needed, and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Information and Regulatory Affairs, Office of Management and Budget, Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE  April 1997		3. REPORT TYPE AND DATES COVERED  Contractor Report, Feb 97
4. TITLE AND SUBTITLE  The ASAC Flight Segment and Network Cost Models			5. FUNDING NUMBERS  C NAS2-14361 WU 538-08-11-01	
6. AUTHOR(S)  Bruce J Kaplan, David A. Lee, Nusrat Retina, and Earl R. Wingrove III, Logistics Management Institute, McLean, Virginia; Brett Malone, Stephen G. Hall, and Scott A. Houser, Phoenix Integration, Blacksburg, Virginia				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Logistics Management Institute 2000 Corporate Ridge McLean, VA 22102-7805			8. PERFORMING ORGANIZATION REPORT NUMBER  LMI- NS602T2	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  NASA CR-201679	
11. SUPPLEMENTARY NOTES Langley Technical Monitor. Robert Yackovetsky Final Report				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Unclassified-Unlimited Subject Category 01 Availability: NASA CASI, (301) 621-0390			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  To assist NASA in identifying research area with the greatest potential for improving the air transportation system, two models were developed as part of its Aviation System Analysis Capability (ASAC). The ASAC Flight Segment Cost Model (FSCM) is used to predict aircraft trajectories, resource consumption, and variable operating costs for one or more flight segments. The Network Cost Model can either summarize the costs for a network of flight segments processed by the FSCM or can be used to independently estimate the variable operating costs of flying a fleet of equipment given the number of departures and average flight stage lengths.				
14. SUBJECT TERMS  Aviation economic models, flight trajectories, airline economic models, aviation system analysis capability. NASA advanced subsonic technology program.			15. NUMBER OF PAGES  198	
			16. PRICE CODE  A09	
17. SECURITY CLASSIFICATION OF REPORT  Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE  Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT  UL	